Fachbereich III Informations- und Kommunikationswissenschaften

Institut für Angewandte Sprachwissenschaft

MAGISTERARBEIT

INTERNATIONALES INFORMATIONSMANAGEMENT

Maschinelles Lernen von Ontologien im Semantic Web im Rahmen des MyShelf-Projekts

vorgelegt von

Daniel Harbig

Erstgutachter: Dr. René Schneider

Zweitgutachter: Dr. Thomas Mandl

Hildesheim, im März 2005

Zusammenfassung

Die vorliegende Arbeit befasst sich mit maschinellem Lernen von Ontologien. Es werden verschiedene Ansätze zum Ontology Learning vorgestellt und diskutiert. Der Fokus liegt auf dem Einsatz maschineller Lernalgorithmen zum automatischen Erwerb von Ontologien für das virtuelle Bibliotheksregal MyShelf. Dieses bietet Benutzern bei der Recherche durch Ontology Switching einen flexibleren Zugang zu Informationsbeständen. Da Ontologien einen Grundbaustein des Semantic Web darstellen, bietet maschinelles Lernen die Möglichkeit, Verfahren zur automatischen Generierung und Verarbeitung von Ontologien zu etablieren. Basierend auf Textkorpora werden Lerntechniken angewandt, um deren Potential für die Erstellung von Ontologien zu überprüfen.

Schlüsselbegriffe:
Ontologie, Maschinelles Lernen, Ontology Learning, Virtuelles Bibliotheksregal, MyShelf, Semantic Web.

Abstract

The goal of this thesis is the introduction of different ontology learning approaches. Different machine learning techniques will be applied focussing on automatic acquisition of ontologies regarding the MyShelf project. MyShelf offers users a flexible way of browsing information by ontology switching. As ontologies are used within the Semantic Web, ontology learning approaches offer opportunities to establish automatic acquisition and processing of ontologies. On the basis of text corpora different ontology learning approaches are applied, in order to test the application of machine learning techniques for the development of ontologies.

Keywords:

ontology, machine learning, ontology learning, virtual shelf, MyShelf, Semantic Web.

Inhaltsverzeichnis

0 Einleitung

„We are drowning in information and starved for knowledge.“

JOHN NAISBETT (1982)

Motivation

Bei der Suche nach Informationen gewinnen digitale Medien und das Internet zunehmend an Bedeutung. Das Internet stellt eine sehr große Bandbreite an Daten zur Verfügung, die potentiell für den Nutzer relevant sein können. Die zunehmende Erweiterung der Recherche auf digitalisierte Medien erfordert die Umsetzung und Bereitstellung neuer Möglichkeiten des Zugriffs auf die darin enthaltenen Informationen. Allerdings steigt die Menge an Informationen rapide, so dass eine Flut an Daten zu bewältigen ist. Wünschenswert wäre weiterhin die Möglichkeit eines Zugriffs auf Informationen aufgrund semantischer Zusammenhänge.

Am Beispiel des Internet stellt sich die Frage, wie die enthaltenen Informationen und folglich das darin enthaltene Wissen zugänglich gemacht werden können, so dass Benutzer auf relevante Informationen strukturiert zugreifen können. Dies führt direkt zum Grundgedanken des Semantic Web. Basierend auf der Idee von Tim Berners-Lee wird im Semantic Web das in den Daten enthaltene Wissen über Ontologien strukturiert und formalisiert, so dass dadurch auch semantische Beziehungen zwischen den Informationen berücksichtigt werden. In Ontologien soll Wissen formal repräsentiert werden, wobei spezifisches Domänenwissen die Grundlage zu deren Erstellung ist. Wie es John Naisbitt im obigen Zitat schildert, steht trotz der Menge an Informationen letztlich das darin enthaltene Wissen im Mittelpunkt. Da die Mengen an Daten zunehmend größer werden und die Aufgabe für den Menschen, diese zu strukturieren und das implizierte Wissen zugänglich zu machen, mit erheblichem Aufwand verbunden ist, müssen Wege gefunden werden, um diese Datenflut bewältigen zu können. Hierbei stellen maschinelle Lernverfahren eine Möglichkeit dar. Mit Hilfe maschinellen Lernens können Aufgaben bei der Datenanalyse an Computer delegiert werden. Somit könnte

die Datenflut überwunden werden. Allerdings stellt sich die Frage, inwieweit die Erstellung von Ontologien automatisiert erfolgen kann, um an das implizierte Wissen gelangen zu können.

Eine Anwendungsmöglichkeit bietet das virtuelle Bibliotheksregal MyShelf. Es stellt die Möglichkeit dar, auf einen Medienbestand über unterschiedliche Ontologien zuzugreifen. Das so genannte Ontology Switching wird zur Überwindung der semantischen Heterogenität von Beständen eingesetzt und bietet die Möglichkeit, unterschiedliche Perspektiven auf Bestände auszuwählen, so dass je nach Benutzerpräferenz bei der Informationssuche eine Perspektive gewählt werden kann. MyShelf wurde für den Bücherbestand der Universitätsbibliothek Hildesheim entwickelt. Aufgrund der zunehmenden Bedeutung digitaler Medien und des daraus resultierenden Benutzerverhaltens bei der Suche sollen Perspektiven auf digitale Medien ermöglicht werden. Da die Erstellung der Perspektiven aufgrund der großen Menge an elektronischern Dokumenten einen sehr großen manuellen Aufwand darstellt, soll untersucht werden, inwieweit maschinelle Lernverfahren zur Erstellung von Ontologien geeignet sind.

Zielsetzung der Arbeit

Ziel dieser Arbeit ist es, diverse Ansätze zum maschinellen Lernen von Ontologien vorzustellen und auf ihre Effizienz zu prüfen. In einem ersten Schritt werden verschiedene Ansätze zum Ontology Learning beschrieben. Anhand eigens erschlossener Textkorpora wird dann versucht, Ontologien maschinell lernen zu lassen. Mit Hilfe verschiedener Lernsysteme werden Versuche zum maschinellen Lernen von Ontologien durchgeführt. Nach einer Evaluierung der Lernergebnisse wird das Potential maschineller Lerntechniken zum Lernen von Ontologien beurteilt. Ferner wird die Anwendung maschinellen Lernens auf Ontologien für das virtuelle Bibliotheksregal MyShelf übertragen.

Aufbau der Arbeit

Die Arbeit befasst sich mit maschinellem Lernen von Ontologien. Dabei wird zunächst in **Kapitel 1** der Rahmen dieser Arbeit abgesteckt und ihr Bezug zu MyShelf erläutert. Hierbei

werden bereits existierende Magisterarbeiten, die sich mit MyShelf befasst haben, vorgestellt. Auch wird die Möglichkeit des Ontology Switching näher betrachtet, sowie der automatische Erwerb weiterer Ontologien hierfür als Grundlage und Ziel dieser Arbeit erläutert. **Kapitel 2** befasst sich mit Ontologien im Rahmen des Semantic Web. Dabei wird das Konzept des Semantic Web erläutert und der Einsatz von Ontologien in diesem beschrieben. Anschließend wird auf Bestandteile und die Vorgehensweise bei der Erstellung von Ontologien eingegangen. **Kapitel 3** stellt die Grundlagen des maschinellen Lernens vor. Anhand verschiedener Definitionen sollen Grundlagen und Anwendungsgebiete vermittelt werden.

In **Kapitel 4** werden Ontology Learning definiert und die Einsatzbereiche vorgestellt. Im Anschluss daran erfolgt die Darstellung verschiedener Lernansätze zum Ontology Learning. Dabei werden allgemeine Ansätze vorgestellt, sowie diejenigen, die bei der Versuchsdurchführung verwendet werden.

Kapitel 5 beschreibt die Vorgehensweise und Anforderungen bei der Durchführung der Lernversuche. Hierbei werden die verwendeten Lernsysteme und die Ausgangsbasis der Versuchsanordnungen vorgestellt und erläutert. Die Ergebnisse der Lernversuche werden in **Kapitel 6** beschrieben und anschließend evaluiert. Die Arbeit schließt mit der Zusammenfassung und dem Ausblick in **Kapitel 7**.

1 MyShelf - Das virtuelle Bibliotheksregal

Im folgenden Kapitel wird das virtuelle Bibliotheksregal MyShelf vorgestellt. Dabei werden unter anderem die Einsatzmöglichkeiten von MyShelf und die darauf basierenden Weiterentwicklungen beschrieben.

1.1 Was ist MyShelf?

Der Gang in eine Bibliothek gehört nach wie vor zu den gebräuchlichsten Methoden bei der Informationssuche. Aufgrund der zunehmenden Fülle an zur Verfügung stehenden Daten im Internet reicht es allerdings nicht mehr aus, bei der Suche nach Informationen nur den Literaturbestand einer Präsenzbibliothek zu berücksichtigen. Um elektronische Dokumente, die mehr und mehr an Bedeutung bei der Recherche gewinnen, auf eine gleichwertige Art und Weise wie den Literaturbestand einer oder sogar mehrerer Bibliotheken einbeziehen zu können, treten diverse Schwierigkeiten auf, diese Dokumentkollektionen dem Benutzer kombiniert zugänglich zu machen. Dabei steht die Möglichkeit im Vordergrund, die Recherche über einen Browsingzugang zu ermöglichen, so dass Benutzer den Bestand nach semantischen Kriterien durchsuchen können. Schwierigkeiten werden durch die semantische Heterogenität hervorgerufen. Diese führt dazu, dass andere Medientypen (elektronische Dokumente aus dem Internet, Audio- und Videoformate) als Bücher in einer bestehenden Systematik eines Literaturbestandes semantisch nicht ausreichend beschrieben werden können, da die Beschreibung des Bestandes auf dessen Sinnzusammenhänge angepasst wurde. Da aber diese Quellen (elektronische Medien) bei der Recherche durchaus von Bedeutung sein können und zunehmend wichtiger werden, muss eine Lösung gefunden werden, diese zu integrieren. Die Angleichung der Systematik an diese Anforderungen stellt allerdings einen großen Aufwand dar, da die Konsistenz einer Systematik aufrechterhalten werden muss.

Der Browsingzugang zu semantisch heterogenen Dokumentkollektionen soll durch ein Modell namens MyShelf ermöglicht werden, welches durch die Virtualisierung des Bestandes die Problematik der Heterogenität löst. Als Metapher dient die Aufstellungssystematik einer Bibliothek, also eines typischen semantischen Ordnungssystems [vgl. MANDL; WOMSER-

HACKER 2002]. Das grundlegende Ziel des virtuellen Bibliotheksregals MyShelf beschreiben MANDL & WOMSER-HACKER (2002) folgendermaßen:

> *„MyShelf zielt darauf ab, die relevanten Bibliotheksbestände, Bestände anderer Bibliotheken, von Dozenten erstellte Lehrmaterialien und Quellen aus dem Internet zu integrieren. Der Zugang umfasst mehrere hierarchische Ordnungssystematiken, wobei vor allem bestehende Bibliothekssystematiken integriert werden. Die Systematik soll vom Benutzer ausgewählt werden können, woraufhin sich die Bücher nach der gewählten Systematik neu anordnen.“*

Für die Universitätsbibliothek (UB) Hildesheim findet dieses Konzept im Bereich Angewandte Informationswissenschaft Anwendung, da hier informationswissenschaftliche Bücher über keine eigene Systemstelle in der Gesamtsystematik der UB Hildesheim verfügen, was darauf zurückzuführen ist, dass die Disziplin Informationswissenschaft der Universität Hildesheim erst spät eingerichtet wurde. Dies hat zur Folge, dass informationswissenschaftliche Literatur aufgrund der Interdisziplinarität des Faches über sehr viele Sachgebiete und somit über sehr viele Signaturen verteilt ist (z.B. Informatik, Linguistik etc.). Auch relevante elektronische Dokumente sowie Quellen aus dem Internet sind nicht nach bibliothekarischen Gesichtspunkten erschlossen. Das Browsing des Bestandes nach informationswissenschaftlichen Titeln ist demzufolge weder am Bücherregal noch im OPAC (Online Public Access Catalogue) der UB Hildesheim möglich. Demzufolge erleichtert eine virtuelle Aufbereitung der bibliothekarischen Inhalte den Zugriff auf diese Titel unter anderem für Studierende der Informationswissenschaft [vgl. HANKE 2002:1].

MyShelf ermöglicht es, die Systematik so zu wechseln, dass „[…] der Mehrwert [dabei] […] ist, dass die gesamte Kollektion immer nach der aktuell ausgewählten Systematik geordnet wird“ [vgl. HANKE ET AL. 2002:296]. Dies bedeutet, dass ein Zugriff auf relevante Quellen aus dem Internet auch über die Systematik der UB Hildesheim stattfinden könnte, oder über eine beliebig andere Systematik, je nachdem, welche Perspektive der Benutzer bevorzugt. Somit könnten auch Abstimmungen auf die Entwicklung und Veränderungen dieser wissenschaftlichen Disziplin schneller erfolgen, da nur die virtuellen Inhalte angeglichen werden müssten [vgl. HANKE 2002:80]. Dieser Wechsel der Perspektive auf einen Bestand nennt man Ontology Switching. Ontologien sind „[…] formale Modelle einer Anwendungsdomäne, die dazu dienen den Austausch und das Teilen von Wissen zu erleichtern“ [vgl. MAEDCHE ET AL. 2001:393]. Der Vorteil von Ontology Switching besteht darin, dass „[h]inter jeder derartigen Einteilung der Welt in Begriffe – sei es in Form eines Thesaurus, einer Klassifikation oder einer sonstigen Ontologie – […] eine individuelle

Sichtweise [steht]“ [vgl. MANDL; WOMSER-HACKER 2002]. Somit ist MyShelf „die Metapher für das Ontology Switching, „[…] das seine Dokumente je nach Perspektive des Benutzers (also der gewählten Ontologie) neu anordnet“ [vgl. KÖLLE ET AL. 2004]. Durch die Integration mehrere Ontologien kann die semantische Heterogenität abgefangen werden [vgl. KÖLLE ET AL. 2004].

Die genaueren Definitionen und Einsatzbereiche von Ontologien werden in Kapitel 2 näher erläutert.

1.2 Umsetzung und Weiterentwicklung von MyShelf

Mehrere Magisterarbeiten haben sich mit dem virtuellen Bibliotheksregal beschäftigt. Die Umsetzung von MyShelf wird im Folgenden näher beschrieben.

Im Rahmen der Magisterarbeit von Peter Hanke „Neue Chancen und Möglichkeiten für Ordnungssystematiken durch Virtualisierung: Anwendung am Beispiel der Erfassung und Klassifizierung des informationswissenschaftlichen Bücherbestandes der Universitätsbibliothek Hildesheim“ [HANKE 2002] wurde aufgrund der bereits erläuterten Heterogenitätsproblematik für den genannten Bücherbestand eine eigenständige Systematik entwickelt, die es ermöglichen sollte, informationswissenschaftliche Titel der UB Hildesheim unter einer eigenen Klassifikation zusammenzufassen.

Um Stärken und Vorteile bestehender Systematiken auszunutzen, wurden als Vorlage die informationswissenschaftlichen Klassifikationen von Bibliotheken benutzt, die über eine eigene Klassifikation informationswissenschaftlicher Themen und Lehrgebiete verfügen. Durch Zuweisung informationswissenschaftlicher Titel der UB Hildesheim zu den Vergleichssystematiken konnte eine eigenständige Klassifikation entworfen und der ermittelte informationswissenschaftliche Literaturbestand der UB Hildesheim anschließend in diese neue Klassifikation eingeordnet werden [vgl. HANKE 2002:81]. So entstand eine Klassifikation mit 16 Kategorien und mehreren Unterkategorien, die des Weiteren als Hanke-Klassifikation bezeichnet wird (siehe Hanke Klassifikation im Anhang I).

Die Hanke-Klassifikation mitsamt den zugeordneten Titeln stellt allerdings nur eine Momentaufnahme dar, da informationswissenschaftliche Bücher, die neu in die UB Hildesheim eingestellt werden, in dieser Systematik nicht berücksichtigt werden und somit nachträglich aufgenommen werden müssen [VGL. HANKE 2002:98]. Peter Hanke entwickelte darüber hinaus einen ersten HTML-basierten Browsingzugang zu den von ihm als relevant identifizierten Büchern. Die einzelnen Systemstellen der Klassifikation wurden mit denen anderer informationswissenschaftlicher Klassifikationen verlinkt, so dass auch das Angebot anderer Standorte nutzbar gemacht werden konnte und umgekehrt.

Eine von HANKE (2002) vorgeschlagene Implementierung eines verbesserten Zugriffs auf die neue Systematik und die dahinter stehenden Bücher wurde von HEINZ (2003) in der Magisterarbeit „Realisierung und Evaluierung eines virtuellen Bibliotheksregals für die Informationswissenschaft an der Universitätsbibliothek Hildesheim“ umgesetzt. Dabei wurde unter anderem eine eigene Web-Benutzeroberfläche[1] auf Hypertextbasis (HTML) implementiert [vgl. HANKE 2002:101], als auch die Systematik erweitert, so dass es nun drei Möglichkeiten für einen Zugriff auf die informationswissenschaftlichen Titel über die Internetseite gibt (siehe Abbildung 1). Benutzer haben die Möglichkeit eine von drei verschiedenen hierarchischen Ordnungssystematiken auszuwählen, nach der sich der Bestand entsprechend anordnet. Bei den drei Systematiken handelt es sich um die Hildesheimer Aufstellungssystematik (Systematik A), die von Hanke entwickelte informations-wissenschaftliche Systematik (Systematik B) und die informationswissenschaftliche Systematik kid (Kybernetik, Informatik, Datenverarbeitung und Informationswissenschaft) der UB Konstanz (Systematik C) [vgl. HEINZ 2003:42].

[1] Siehe [HEINZ 2003a].

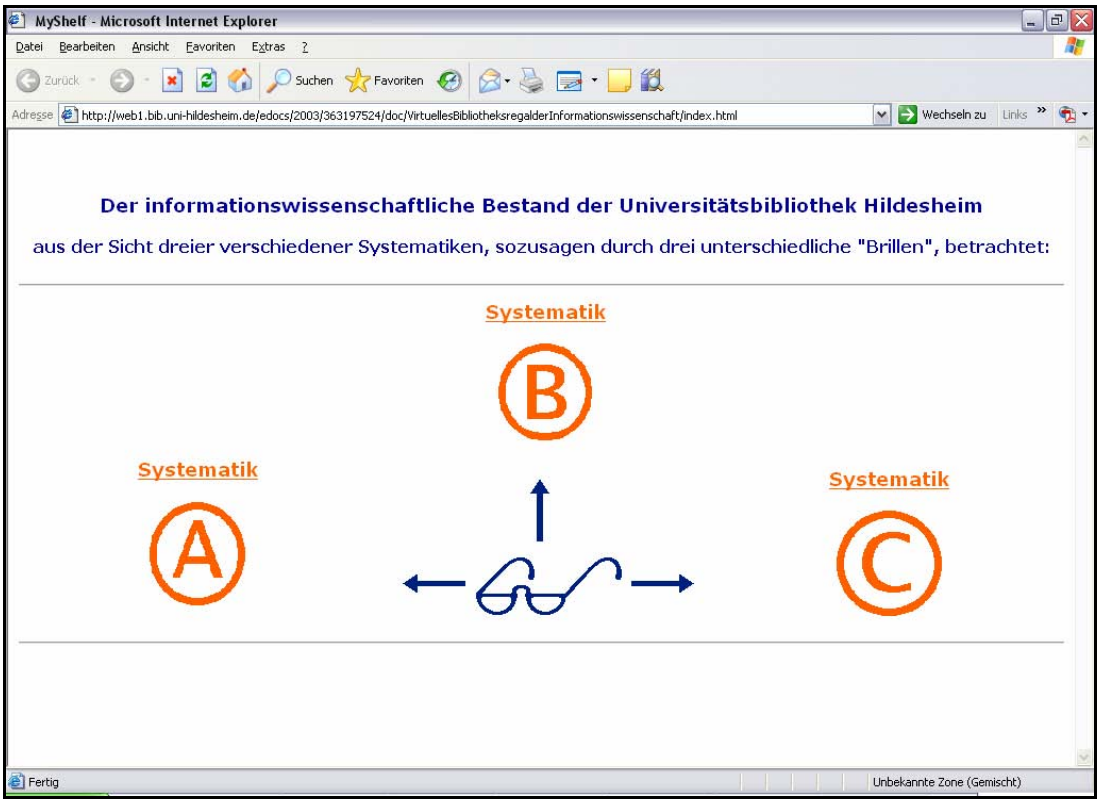


Abbildung 1: Benutzeroberfläche mit drei Systematiken [HEINZ 2003a]

Dieser Einsatz verschiedener Perspektiven wurde bereits weiter oben geschildert und wird als Ontology Switching bezeichnet.

Basierend auf der Arbeit von HANKE (2002) und HEINZ (2003) entwickelte WILHELM (2004) einen erweiterten Browsingzugriff auf die bereits erschlossene Literatur, indem sie den Literaturbestand der UB Hildesheim um eine bewertete Linksammlung erweiterte. Ziel der Magisterarbeit „Der virtuelle Wegweiser Informationswissenschaft - Entwicklung und Implementierung eines Konzepts für die Integration eines Clearinghouse in das virtuelle Bibliotheksregal MyShelf“ [WILHELM 2004] war die Entwicklung eines Clearinghouse Virtueller Wegweiser Informationswissenschaft, in welchem „Internetressourcen zu einem bestimmten Fachgebiet nicht nur gesammelt, sondern auch strukturiert [werden], so dass sie eine Schnittstelle zwischen dem Benutzer und den im Internet bereitstehenden Informationen darstellen“ [VGL. WILHELM 2004:1]. Die Auswahl der Internetquellen erfolgte intellektuell und wurde nach einem eigens entwickelten Bewertungssystem durchgeführt, wobei die Links aus den informationswissenschaftlichen Teilgebieten Human-Computer-Interaction und Sprachtechnologie erschlossen wurden [VGL. WILHELM 2004:2].

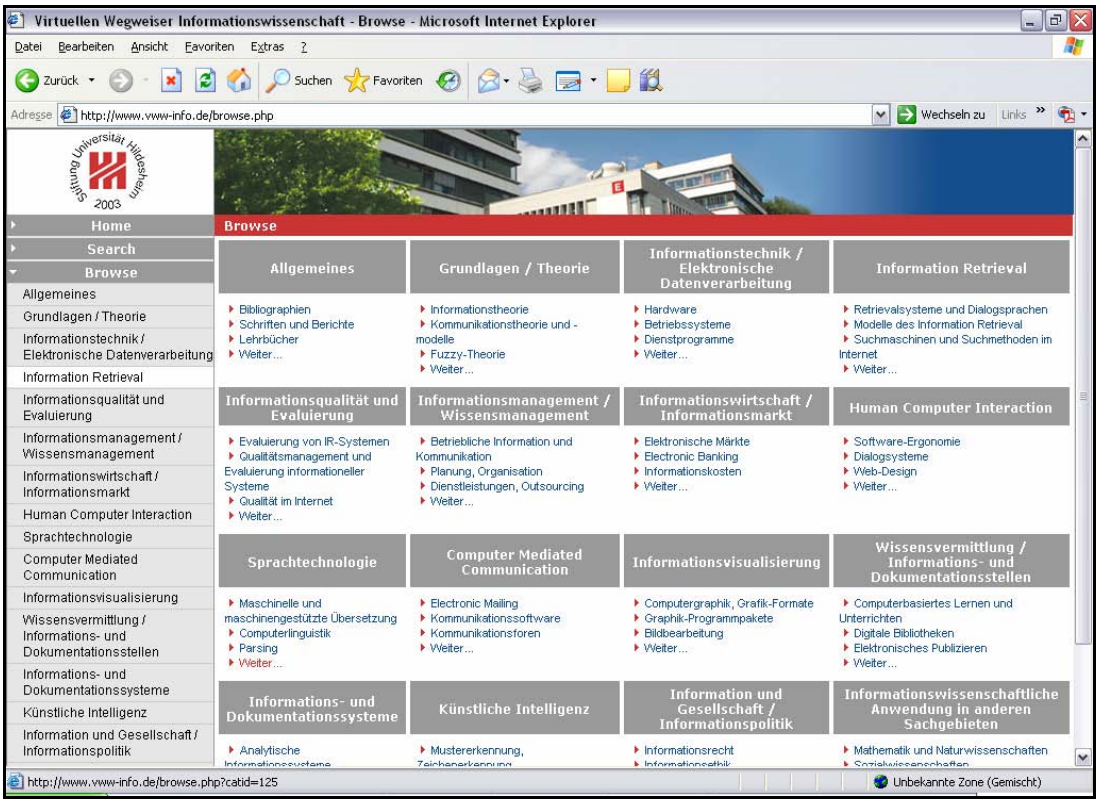


Abbildung 2: Der Virtuelle Wegweiser Informationswissenschaft [WILHELM 2004a]

Die Linksammlung wurde auf Basis der Hanke-Klassifikation für die Bücher der Universitätsbibliothek Hildesheim strukturiert. Der Unterschied zu der Arbeit von HEINZ (2003) ist, dass hierbei nur ein Klassifikationssystem verwendet wird und kein Wechsel der Perspektiven auf den Gesamtbestand möglich ist (siehe Abbildung 2) [VGL. WILHELM 2004:27f].

Des Weiteren wurde in der Magisterarbeit von Svenja Wiegemann „Implementierung einer benutzungsfreundlichen Oberfläche für mobile Endgeräte am Beispiel eines Bibliotheksinformationssystems“ [Wiegemann 2004] die Entwicklung und Implementierung eines Prototypen für mobile Endgeräte vorgestellt, der den Zugriff auf ein Bibliotheksinformationssystem ermöglicht. Hierbei wurde MyShelf als Ausgangsbasis zur Umsetzung dieser Magisterarbeit verwendet, auf welche an dieser Stelle nicht weiter eingegangen werden soll.

1.3 Erweiterung des Ontology Switching

Das MyShelf-Konzept konnte in den vorher besprochenen Arbeiten umgesetzt und weiterentwickelt werden. Wie bereits beschrieben, ist durch Ontology Switching in MyShelf die Möglichkeit gegeben, die semantische Heterogenität abzufangen. HEINZ (2003) hat bereits drei Perspektiven implementiert, die je nach Bevorzugung vom Benutzer gewechselt werden können. Um diese Möglichkeit des Perspektivenwechsels auch auf andere Bestände erweitern zu können, soll in dieser Arbeit im Rahmen von MyShelf untersucht werden, inwieweit Ontologien, die als Perspektiven eingesetzt werden sollen, maschinell gelernt werden können. Da der manuelle Aufwand zur Erstellung einer derartigen Ontologie sehr groß ist, bieten maschinelle Lernverfahren eine Möglichkeit, diesen Aufwand zu minimieren oder sogar zu beseitigen. Bei dieser Untersuchung stehen keine Literaturbestände im Mittelpunkt, sondern die Integration von Quellen aus dem Internet, die zunehmend Bedeutung bei der Recherche gewinnen. WILHELM (2004) hat den Hanke-Klassen bereits diverse Internetsites zugeordnet.

Auf dieser Basis soll im Laufe dieser Arbeit der Einsatz maschineller Lernverfahren zur Erstellung von Ontologien untersucht werden. Dadurch könnten Ontologien für semantisch heterogene Quellen aus dem Internet erstellt werden, so dass hierfür kein Systemwechsel nötig wäre, da alles für den Suchenden relevante Wissen bereits in der Ontologie enthalten wäre. Somit soll erreicht werden, dass elektronische Dokumente auf eine gleichwertige Art und Weise wie ein Literaturbestand einer oder sogar mehrerer Bibliotheken über MyShelf zugänglich gemacht werden bzw. verschiedene Perspektiven auf Bestände ermöglicht werden.

In diesem Kapitel wurde MyShelf vorgestellt und der Einsatz maschineller Lernverfahren hierfür bzw. für das Ontology Switching wurde erläutert. In den nächsten Kapiteln sollen Ontologien und deren Einsatz im Semantic Web näher behandelt werden.

2 Semantic Web und Ontologien

"Tell me what wines I should buy to serve with each course of the following menu. And, by the way, I don't like Sauternes."

zitiert nach W3C (2004)

Wenn man dieses Zitat liest, ist anfangs nicht genau klar, an wen es gerichtet ist und wie es in den Verlauf dieser Arbeit passen soll. Jedoch spiegelt dieses Zitat die Anforderungen wider, denen Anwendungen im Semantic Web begegnen könnten, wenn sie Aufträge von Benutzern bearbeiten sollen. Im Folgenden soll eine kurze Einführung den Grundgedanken des Semantic Web erläutern. Auch werden in diesem Zusammenhang Ontologien als Bestandteile des Semantic Web vorgestellt.

2.1 Das intelligente Netz

Das Internet hat sich in den letzten Jahren weitgehend als globales Medium etabliert und weiterentwickelt. Aufgrund seines steten Wachstums, dient es als universelle Quelle und als neue Zugangsmöglichkeit zu Informationen. Diese Universalität und Zugänglichkeit für jedermann machen es jedoch unmöglich, eine flächendeckende Kontrolle der enthaltenen Inhalte zu etablieren, da es unzählige Autoren gibt. So wie wir es heute kennen, kann die im Internet enthaltene Information von Maschinen nicht sinngerecht interpretiert werden; sie wird lediglich angezeigt. Das liegt daran, dass das World Wide Web (WWW) für Menschen konzipiert wurde und deshalb für Maschinen nur bedingt interpretierbar ist. Dieser Herausforderung stellt sich das Semantic Web. Das Ziel ist es, Daten und Informationen für Computer interpretierbar zu machen. Wie der Name schon sagt, befasst sich die Semantik (griech. seman- = bedeuten) mit der Bedeutung bzw. dem Bedeutungspotential von Sprache [vgl. KORTMANN 1999:155]. Bislang transportiert das Web vor allem für Menschen lesbare Informationen, jedoch wurden dabei automatisch verarbeitbare Daten vernachlässigt. Das

semantische Netz soll folglich die Erfassung von Bedeutung von Information im Web für Maschinen ermöglichen.

Die grundlegende Idee stammt von Tim Berners-Lee, der das Semantic Web als eine Erweiterung des heute bekannten Internet Web definiert [BERNERS-LEE ET AL. 2001:31]:

> *"The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation."*

Demzufolge beschreiben MAEDCHE ET AL. (2001:397), dass es sich „beim Semantic Web […] deshalb auch nicht einfach um eine Anwendung [handelt], sondern vielmehr um eine generelle Vision und Architektur für die Entwicklung von webbasierten Anwendungen der nächsten Generation“. Diese Erweiterung soll es maschinellen Akteuren ermöglichen, Daten im Web durch Maschinen besser verarbeiten und semantisch interpretieren zu können. Erreicht man, das Internet für Maschinen verständlicher zu gestalten, wird es letztlich auch für den menschlichen Benutzer nützlicher, da Applikationen zielgerichteter arbeiten können. Betrachtet man zum Beispiel das obige Zitat des W3C, so könnte eine Anwendung für das Semantic Web so aussehen, dass zur Vorbereitung eines mehrgängigen Menüs ein Softwareagent selbstständig passende Weine für die einzelnen Gänge recherchiert und vorschlägt, wobei er speziell die Präferenzen des Benutzers beachtet, so dass bei diesem Beispiel keine Sauterne-Weine berücksichtigt werden sollen. Dies stellt eine recht komplexe Anforderung für den Agenten dar, da aus semantischer Sicht Benutzerpräferenzen richtig interpretiert werden müssen. Das Semantic Web selbst stellt keine Applikationen für derartige Einsatzmöglichkeiten zur Verfügung, worauf MAEDCHE ET AL. (2001:397) bereits hinweisen.

2.2 Architektur und Standards des Semantic Web

Maschinen können die Informationen auf Internetseiten nicht nutzen, da sich das WWW auf die Präsentation von Information beschränkt. Da Maschinen kaum in der Lage sind, Informationen bezüglich eines Kontextes zu unterscheiden, müssen semantische Zusammenhänge explizit von einem menschlichen Benutzer erkannt werden [vgl. KÖLLE ET AL. 2004:113].

Aus diesem Grund bietet das Semantic Web die Möglichkeit, Informationen über Metadaten zu beschreiben und semantische Beziehungen zwischen Informationen und dem vorgesehen Kontext auszudrücken. Um dies zu ermöglichen und um die bereits genannte Universalität des WWW aufrecht zu erhalten, müssen globale Standards definiert werden, damit das Potential des Semantic Web ausgeschöpft werden kann [vgl. BERNERS-LEE ET AL. 2001:43]. Eine passende Beschreibungssprache soll sowohl Daten als auch Schlussregeln ausdrücken können, wodurch es möglich wird, Daten universell in jedes beliebige Expertensystem zu übersetzen oder in das Internet zu transportieren [vgl. BERNERS-LEE ET AL. 2001:32]. Zwei wichtige Bestandteile für dieses Entwicklung sind die erweiterbare Markup-Sprache XML und das Ressourcenbeschreibungssystem RDF. XML (Extensible Markup Language) ist eine Beschreibungssprache mit einer ähnlichen Struktur wie HTML (Hypertext Markup Language). Im Gegensatz zu HTML stellt XML Etiketten (tags) bereit, die vom Benutzer individuell definiert werden können. Somit kann man Inhalte einer Website um diese Tags erweitern, so dass damit Informationen maschinenlesbar gekennzeichnet und zur Verfügung gestellt werden. Zum Beispiel kann ein Tag <PLZ> die Postleitzahlen auf einer Seite eindeutig kennzeichnen, welche ohne diese Kennzeichnung nicht unbedingt als solche zu verstehen wären. XML unterstützt bei der Strukturierung einer Website, jedoch vermittelt sie nicht die Bedeutung der neuen Tags von einem Benutzer zum anderen [vgl. BERNERS-LEE ET AL. 2001:32f]. Diese einfachen inhaltlichen Zusammenhänge werden durch die Sprache RDF (Resource Description Framework) ausgedrückt. RDF stellt ein gemeinsames Format für Metadaten (Informationen über Informationen) dar. Somit können Daten aus allen möglichen Bereichen einheitlich dargestellt werden. Der Aufbau eines RDF-Ausdrucks beinhaltet drei Glieder, worin jedes Glied mit XML-Tags beschrieben werden kann. Es ähnelt der Syntax eines Satzes mit Subjekt, Prädikat und Objekt, so dass jede Beziehung durch ein Tripel dargestellt wird. Somit wird ausgedrückt, dass ein Subjekt (eine Person, eine Website etc.) in einer bestimmten Relation ("ist Schwester von", "ist Autor von") zu einem Objekt (einer anderen Person, Website etc.) steht [vgl. BERNERS-LEE ET AL. 2001:32 ff]. Laut BERNERS-LEE ET AL. (2001:32ff) lässt sich damit ein Großteil der maschinenlesbaren Informationen beschreiben. Die Relationen zwischen Subjekt und Objekt werden durch Prädikate ausgedrückt. Durch diese so genannten RDF-Tripel (Subjekt, Prädikat und Objekt) entsteht ein Netz von Informationen über miteinander in Beziehung stehende Begriffe [vgl. BERNERS-LEE ET AL. 2001:32 ff].

XML und RDF stellen nur einen Teil der Architektur des Semantic Web dar. Dabei enthält der so genannte Layer Cake nach BERNERS-LEE (2000) weitere Schichten, die jeweils eine Erweiterung ihrer vorhergehenden Schicht darstellen (siehe Abbildung 3).

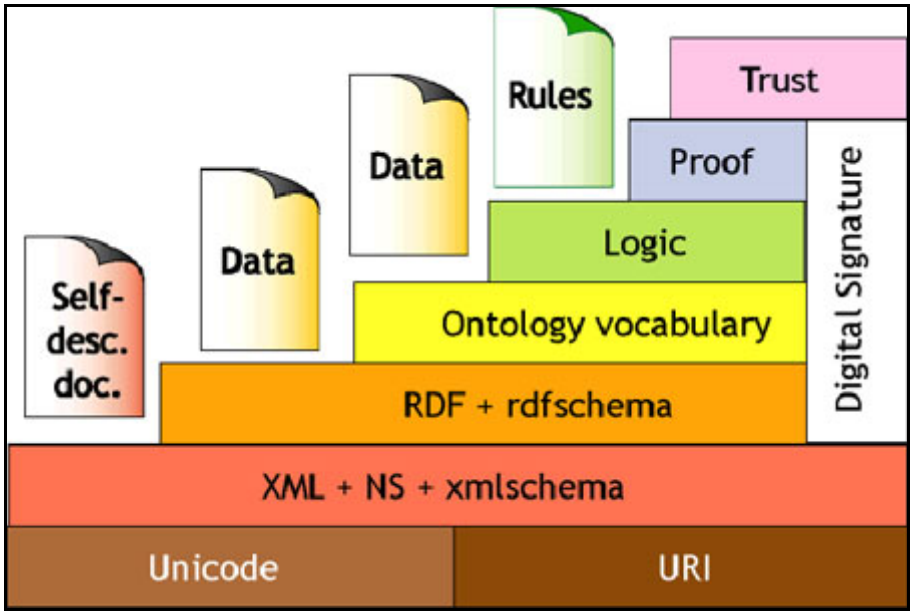


Abbildung 3: Das Schichtenmodell des Semantic Web in BERNERS-LEE (2000).

Ontologien bieten Kollektionen von Informationen und Wissen über eine spezifische Domäne, so dass man bei RDF-Tripel sicher gehen kann, dass sie auf der Basis gleicher Konzepte beruhen. Die gebräuchlichste Art von Ontologien für das WWW sind Taxonomien. Auf die folgenden Schichten Logic, Proof und Trust soll nicht weiter eingegangen werden. An dieser Stelle soll nur kurz daraufhin gewiesen werden, dass sich diese Schichten damit befassen, auf Basis der RDF-Tripel Schlussregeln zu bilden und durch Überprüfungen zu gewährleisten, dass im Semantic Web nur verifizierte Informationen enthalten sind [vgl. LIEBSCH 2004].

Nachdem es unter anderem bei XML jedem Autoren erlaubt ist, eigene Tags zu definieren, oder dieser auch z.B. Datenbankfelder frei festlegen und benennen kann, gibt es im Semantic Web zwangsläufig Probleme ohne eine Vereinheitlichung, da Begriffe von Benutzern oder anderen Autoren unterschiedlich verwendet werden. Die Lösung dieses Problems bieten die bereits genannten Ontologien, die eine Bibliothek von speziellen Informationen einer Wissensdomäne darstellen. Sie bilden die Basis dafür, das Semantic Web für Maschinen interpretierbar zu machen.

2.3 Ontologien

Der Begriff Ontologie stammt ursprünglich aus der Philosophie und bezeichnet die Theorie vom Wesen der Existenz. Wissenschaftlich gesehen geht es um die Frage, welche Typen von Dingen überhaupt existieren. Für die Anwendung im Semantic Web bieten Ontologien die Grundlage zur Beschreibung von Domänenwissen durch die formale Definition von Relationen zwischen Dingen [vgl. BERNERS-LEE ET AL. 2001:34]. Nach MAEDCHE ET AL. (2001:393) sind Ontologien „[…] formale Modelle einer Anwendungsdomäne, die dazu dienen den Austausch und das Teilen von Wissen zu erleichtern“. Eine Domäne ist ein abgeschlossener Wissensbereich einer spezifischen Fachrichtung, in welchem folglich die Fachinformationen in semantischem Kontext stehen. Es gibt mehrere Definitionen für Ontologien, aber nach STAAB & STUDER (2004:VII) hat man sich weitestgehend auf die Definition nach GRUBER (1993) geeinigt:

> *„An ontology is a formal explicit specification of a shared conceptualization for a domain of interest.“*
>
> zitiert nach STAAB & STUDER (2004:VII)

Konzeptualisierung bezieht sich dabei auf ein abstraktes Modell einer Sache oder eines Fachbereiches, welches relevante Konzepte dieses Bereiches identifiziert. Dabei sind die Art des verwendeten Konzepts und der Anwendungskontext, in welchem es gesehen werden muss, *explizit* festgelegt. *Formal* bezeichnet die Eigenschaft, dass Ontologien maschinenlesbar sein soll, wobei unterschiedliche Formalismen möglich sind. Ontologien liefern zudem ein gemeinsames Vokabular an Termen und Beziehungen, mit denen man die Domäne modellieren kann. Dies beinhaltet maschinenlesbare Definitionen von grundlegenden Konzepten einer Domäne und deren Beziehungen (Relationen) zueinander [vgl. NOY; MCGUINESS 2001]. Da das Ziel konsensuelles Domänenwissen ist, also Wissen über eine Fachrichtung, das gemeinsam als spezifisches Domänenwissen anerkannt wurde, ist die Entwicklung einer Ontologie oft ein kooperativer Prozess, der mehrere Leute einbezieht. Dies wird auch durch den Begriff *shared* der Definition deutlich, so dass Wissen nicht nur von einem Individuum festgelegt, sondern von einer Gruppe in Übereinstimmung akzeptiert wird [vgl. FENSEL et al. 2003]. Folglich beziehen sich Domänenontologien immer nur auf eine begrenzte Wissensdomäne und einen begrenzten Personenkreis oder Anwendungsgebiet [vgl. STAAB; STUDER 2004:VII]. Im Bezug auf das Semantic Web bedeutet dies unter anderem,

dass Ontologien semantische Modelle darstellen, die spezifisches Wissen einer Domäne beschreiben, indem sie Daten interpretieren und zueinander in eine bezüglich ihrer Domäne semantisch korrekte Beziehung setzen [vgl. MAEDCHE ET AL. 2001:397]. Dabei werden Ontologien von Benutzern oder Applikationen eingesetzt, die dieses gemeinsame Domänenwissen benötigen. Ontologien ermöglichen es, dass das enthaltene Wissen einer Domäne wiederverwendet werden kann [vgl. W3C 2004a].

2.3.1 Aufbau einer Ontologie

Nach BERNERS-LEE ET AL. (2001:34) besteht eine Web-Ontologie aus einer Taxonomie und einer Liste von Schlussregeln. Eine Taxonomie drückt eine is-a-Beziehung zwischen Konzepten aus, was eine Hierarchie erzeugt. is-a-Beziehungen sagen aus, dass ein Konzept eine Spezialisierung eines anderen Konzeptes ist, also zum Beispiel „Blume“ eine Spezialisierung von „Pflanze“ ist.

Eine Taxonomie definiert Objektklassen und beschreibt die Relationen dieser zueinander. Damit können Unterklassen von ihren Oberklassen alle Eigenschaften erben. So würde eine Taxonomie eine Adresse als eine spezielle Art von Ortsangabe definieren und dabei festlegen, dass eine Postleitzahl sich nur auf eine Ortsangabe beziehen kann etc. [vgl. BERNERS-LEE ET AL. 2001:34]. Durch die hierarchische Struktur stehen Unterklassen in einer transitiven Beziehung zueinander.

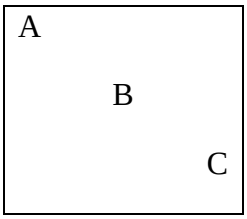


Abbildung 4: Beispiel für Transitivität.

Dies wird in Abbildung 4 deutlich. Hierbei sind durch den unterschiedlichen Einschub die Hierarchieebenen gekennzeichnet. So ist in diesem Fall das Konzept B ein Unterkonzept von A und C ein Unterkonzept von B. Durch die transitive Beziehung ist C auch eine Unterklasse von A und stellt so gesehen eine Spezialisierung von A dar. Wichtig bei der Konzepthierarchie ist es auch, dass es nicht nur einzelne untergeordnete Konzepte pro Klasse geben sollte [vgl. NOY; MCGUINESS 2001]. Dieser Aufbau mit Klassen, Unterklassen und

Relationen ermöglicht automatisiertes logisches Schließen. Eine Ontologie kann somit auch "Wenn-dann“-Regeln ausdrücken. Der Computer "versteht" nichts von alldem im wörtlichen Sinne, kann aber mit den Begriffen auf eine für den Nutzer hilfreiche und einleuchtende Weise umgehen [vgl. BERNERS-LEE ET AL. 2001:34].

Ontologien sollen Konzepte der Domäne, Beziehungen der Klassen untereinander und Eigenschaften der Konzepte beinhalten. NOY & MCGUINESS (2001) beschreiben die grundlegende Vorgehensweise zur Erstellung einer Ontologie sowie Regeln, die dabei einzuhalten sind. Grundbestandteile einer Ontologie sind demnach:

- Konzepte („concepts“ oder „classes“),
- Konzepthierarchie („concept hierarchy“ oder „class hierarchy“),
- Konzepteigenschaften („properties“ oder „slots“),
- Wertebereiche der Konzepteigenschaften („facets“),
- und Instanzen („instances“).

Somit bildet eine Konzepthierarchie zusammen mit einzelnen Instanzen dieser Konzepte innerhalb einer Ontologie eine domänen-spezifische Wissensbasis. Die einzelnen Schritte zur Erstellung einer Ontologie können nach NOY & MCGUINESS (2001) folgendermaßen aussehen:

1. Bestimmung der Domäne und Anwendung
2. Nutzung bestehender Ontologien
3. Sammlung wichtiger Bestandteile der Ontologie
4. Definition der Konzepte und der Konzepthierarchie
5. Definition der Konzepteigenschaften
6. Definition der Wertebereiche
7. Erzeugung von Instanzen

Auch bei der Erarbeitung der Konzepthierarchie gibt es unterschiedliche Vorgehensweisen. Beim Top-Down-Ansatz beginnt man mit der allgemeinsten Klasse und erschließt daraufhin die spezifischeren Konzepte. Genau anders herum verhält es sich beim Bottom-Up-Ansatz, wo man mit der spezifischsten Klasse beginnt und man sich schrittweise zu den

Generalisierungen nach oben arbeitet. Auch gibt es eine hybride Form, die beide genannte Ansätze kombiniert, so dass erst die mehr umfassenden Konzepte definiert und dann entweder Verallgemeinerungen oder Spezifizierungen vorgenommen werden [vgl. NOY; MCGUINESS 2001]. NOY & MCGUINESS (2001) schildern, dass die Reihenfolge dieser Schritte nicht immer zwangsläufig aufeinander folgen muss, und dass die Vorgehensweise von dem Anwendungskontext abhängt, so dass es keine generell richtige Vorgehensweise bei der Ontologieerstellung gibt.

Zusammengefasst beinhaltet die Erstellung der Ontologie also die Definition von Klassen, die Anordnung der Klassen in einer Hierarchie, Festlegung der Eigenschaften und der dafür vorgesehenen Werte, als auch das Hinzufügen von Instanzobjekten zu ihren jeweiligen Konzepten [vgl. NOY; MCGUINESS 2001].

2.3.2 Formalisierung von Ontologien

Zur Formalisierung von Konzepten und Beziehungen innerhalb einer Ontologie bedarf es Ontologiesprachen. Diese sollen mit Hilfe von formalen Semantiken präzise die Bedeutung von Wissen innerhalb der Domäne beschreiben. Dabei dürfen diese Semantiken weder auf subjektiven Intuitionen beruhen, noch dürfen sie von anderen Leuten anders ausgelegt werden [vgl. ANTONIOU; VAN HARMELEN 2003:68ff]. Ontologien sollten daher in auf Logik basierenden Sprachen ausgedrückt werden, so dass detaillierte, akkurate, konsistente und bedeutungsvolle Unterscheidungen zwischen den Klassen, Eigenschaften und Beziehungen gemacht werden können. Einige Tools können automatisierte Schlussfolgerungen durchführen, indem sie die Ontologien benutzen. Somit wird die Möglichkeit geschaffen, intelligente Applikationen (semantische Suche und Retrieval, intelligente Agenten, Entscheidungsunterstützung) einzusetzen [vgl. W3C 2004a].

Man hat bereits die Notwendigkeit für eine mächtigere Ontologie-Modellierungssprache erkannt. Dies führte zu DAML+OIL (zusammengesetzt aus dem amerikanischen Vorschlag DAML-ONT und dem europäischen Sprache OIL), die auf RDF und RDFS aufgebaut sind. Von DAML+OIL ausgehend wurde von der W3C Arbeitsgruppe die Ontologiesprache OWL (Web Ontology Language) definiert, eine Sprache mit dem Ziel einen weit akzeptierten Standard als Ontologiesprache für das Semantic Web zu erfüllen [vgl. ANTONIOU; VAN

HARMELEN 2003:68]. Es gibt mehrere Arten von OWL: OWL Lite, OWL DL (Description Logic) und OWL FULL auf die jedoch an dieser Stelle nicht weiter eingegangen wird.

In diesem Kapitel wurden Ontologien vorgestellt und die Notwendigkeit und Wichtigkeit ihres Einsatzes im Semantic Web erläutert. Auf dieser Basis soll im Laufe dieser Arbeit der Einsatz maschineller Lernverfahren zur Erstellung von Ontologien untersucht werden.

3 Grundlagen maschinellen Lernens

In den folgenden Abschnitten werden Grundlagen maschinellen Lernens vorgestellt. Basierend auf maschinellem Lernen, werden Grundlagen sowie diverse Lernansätze besprochen. Die Betrachtung dieser Ansätze soll die direkte Verbindung zum maschinellen Lernen von Ontologien darstellen.

3.1 Wissen aus Mustern entdecken

"It has been estimated that the amount of information in the world doubles every 20 months."

zitiert nach FRAWLEY ET AL. (1992:57)

Aufgrund der zunehmenden Möglichkeiten, Daten und Informationen über Menschen und Dinge zu speichern, und der daraus resultierenden Menge kommt es zu einer regelrechten Datenflut. Dabei sollen zum Beispiel Daten aus den verschiedensten Bereichen der Wirtschaft, Wissenschaft, Industrie oder des WWW zur Auswertung verwendet werden. Nach FRAWLEY ET AL. (1992) verdoppelt sich die Anzahl der Informationen alle 20 Monate. Da eine rein manuelle Verarbeitung und Analyse der Daten aufgrund dieser enormen Menge sehr zeitaufwendig bzw. nicht durchführbar ist, bedarf es maschineller Unterstützung, um all diese gespeicherten Informationen auswerten zu können. Aufgrund der erheblichen Größe von Datensätzen ist eine maschinelle Verarbeitung ein willkommener Weg zur Delegation von Aufgaben an Computer, da somit wichtige manuelle Arbeitschritte abgelöst werden können und der Zeitaufwand wesentlich reduziert wird. Mit der Unterstützung und Auswertung der Daten befasst sich das Data Mining.

Um aus der Datenflut Wissen ableiten zu können, sollen Muster entdeckt und daraus Wissen abgeleitet werden [vgl. FAYYAD ET AL. 1996:37ff]:

> *"Data mining is the application of specific algorithms for extracting patterns from data. […]. The additional steps in the KDD [Knowledge Discovery in Databases] process, such as data preparation, data selection, data cleaning, incorporation of appropriate prior knowledge, and proper interpretation of the results of mining, are essential to ensure that useful knowledge is derived from the data."*
>
> [FAYYAD ET AL. 1996:39]

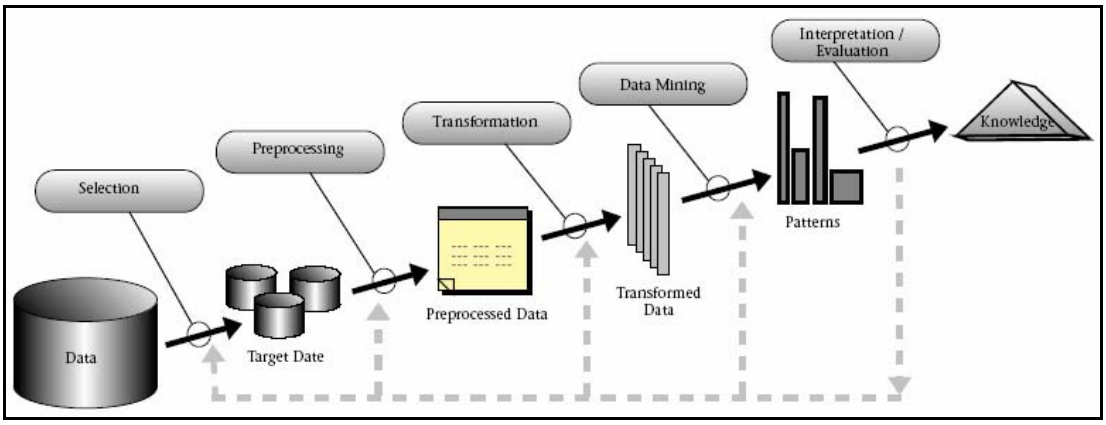


Abbildung 5: Der Prozess der Wissensentdeckung in FAYYAD ET AL. (1996:41).

WITTEN ET AL. (2001) beschäftigen sich mit maschinellem Lernen als Anwendung für Data Mining. Sie definieren Data Mining als „ […] Prozess, Muster in Daten zu erkennen. Der Prozess hat automatisch oder (was häufiger der Fall ist) halbautomatisch stattzufinden. Dabei müssen sinnvolle Muster erkannt werden, die zu einem Vorteil in der Regel wirtschaftlicher Art führen" [WITTEN ET AL. 2001:3]. Damit wird eine Brücke von der Sammlung von Daten bis hin zur ihrer Interpretation dieser geschlagen, da wichtige Informationen in ihnen enthalten sein können, die aber nicht explizit zugänglich sind [vgl. WITTEN ET AL. 2001:2]. Es ist deswegen wichtig Muster zu entdecken, weil man auf deren Basis nicht-triviale Vorhersagen im Hinblick auf Datensätze treffen kann, was bei der Wissensentdeckung eine zentrale Rolle spielt [vgl. WITTEN ET AL. 2001:3]. Muster sind hierbei wiederholt auftretende Datenstrukturen, die Merkmale von Datensätzen repräsentieren. Der Prozess der Wissensentdeckung ist interaktiv und erfolgt in mehreren Teilschritten [vgl. FAYYAD ET AL. 1996:39f]. Wie in Abbildung 5 dargestellt, werden die zu analysierenden Daten ausgewählt und zur maschinellen Weiterverarbeitung aufbereitet und transformiert. Der darauf folgende Data Mining Prozess wird mit Hilfe maschineller Techniken durchgeführt, um Muster in den Daten zu finden. Anschließend werden die Ergebnisse interpretiert und evaluiert, um zu überprüfen, ob durch den Verarbeitungsprozess neues Wissen entstanden ist.

Wie bereits erwähnt, gehört zu den Data Mining Techniken das maschinelle Lernen. Dies kommt immer dann zum Einsatz, wenn manueller Aufwand zu groß oder zu zeitaufwendig erscheint. Dabei sollen Maschinen Regeln selbständig „lernen“, so dass diese für die Verarbeitung von Daten benutzt werden können, ohne eine Interaktion mit einem Benutzer. Maschinelles Lernen ist eine sehr vielschichtige Disziplin, welche Teile aus den Bereichen Künstliche Intelligenz, Statistik, Philosophie, Psychologie, Neurobiologie und vielen mehr vereint. Die Anwendungsbereiche für Maschinelles Lernen sind sehr vielfältig. Somit erstreckt sich der Anwendungsbereich von finanzanalytischen Vorgängen bis hin zur Auswertung von biologischen Daten etc. [vgl. MITCHELL 1997:17].

3.2 Allgemeine Definitionen

Um mehrere Definitionen von maschinellem Lernen zu nennen, soll zunächst geklärt werden, wie Lernen definiert ist. Lernen bedeutet, sich durch Studium, Erfahrung oder Lehre Wissen anzueignen. Dabei wird man sich mittels Informationen oder Beobachtungen einer Sache bewusst [vgl. WITTEN ET AL. 2001:6]. Im Bezug auf Computer ist diese Definition unzulänglich, da es kaum Kontrollmöglichkeiten gibt, ob ein Lernprozess stattgefunden hat oder nicht. WITTEN ET AL. (2001) erklären an dieser Stelle, dass dadurch die Frage aufgeworfen wird, ob ein Rechner Wissen über Dinge besitzen kann und ob er ein Bewusstsein hat. Jedoch soll auf diesen Aspekt nicht weiter eingegangen werden. Bezüglich des maschinellen Lernens sollen Leistungsverbesserungen in neuartigen Situationen im Vordergrund stehen, d.h. dass Lernen bei Maschinen an Leistung gebunden wird und nicht an Wissen. Lernvorgänge, die „unbewusst“ erfolgen, also ohne Absicht, werden als Training bezeichnet [vgl. WITTEN ET AL. 2001:6f]. Ein Computerprogramm lernt also dann, wenn es seine Leistung bei einer zu bewältigenden Aufgabe mit zunehmender Erfahrung verbessert [vgl. MITCHELL 1997:2]. Da es mehrere Definitionen maschinellen Lernens gibt definieren WITTEN ET AL. (2001:6) eine operationale Definition von Lernen:

> *„Etwas lernt, wenn es sein Verhalten so ändert, dass es in Zukunft eine bessere Leistung aufweist.“*

Letztlich werden beide Definitionen des Lernens benutzt: das Aneignen von Wissen sowie die Fähigkeit, es zu nutzen. Bei Data Mining geht es um praktisches Lernen, nicht um theoretisches Lernen.

3.3 Konzepte, Instanzen und Attribute

Unabhängig von den angewandten Lernalgorithmen wird das, was durch ein Lernverfahren gelernt werden soll, als Konzept bezeichnet [vgl. WITTEN ET AL. 2001:42]. Das Ziel des maschinellen Lernens sind also eindeutige und damit verarbeitbare Konzeptbeschreibungen. Dabei liegen die Informationen, die dem Lernverfahren zugeführt werden, in Form einer Instanzenmenge vor. Diese Instanzen (Beispiele bei Eingaben) sind die Dinge, die klassifiziert, assoziiert oder gruppiert werden sollen. Sie sind individuelle, unabhängige Typen eines Beispiels für das Konzept. Dabei stellt jede Instanz ein Beispiel für das zu lernende Konzept dar. In diesen Datensätzen befinden sich keine Hinweise auf die Beziehungen der einzelnen Instanzobjekte zueinander. [vgl. WITTEN ET AL. 2001:42ff]. Die Instanzen werden durch Attribute mit bestimmten Werten charakterisiert [vgl. WITTEN ET AL. 2001:49ff]. Diese Attribute beschreiben unterschiedliche Eigenschaften der Instanzen. Sie sind deswegen sehr wichtig, weil sie die „messbaren“ Aspekte einer Instanz darstellen, also das, was für das Lernen verwendet wird.

Die gesammelten Daten müssen also zur Weiterverarbeitung in einer Form vorhanden sein oder in eine Form gebracht werden, in welcher die Attribute und deren Werte so vorliegen, dass sie genau auf das ausgewählte Lernverfahren abgestimmt sind. Dabei sollen die Daten sinnvoll aggregiert werden und fehlerhafte Datensätze bereinigt werden, bevor sie anschließend in einer Menge von Instanzen zusammengefasst werden können [vgl. WITTEN ET AL. 2001:53ff].

3.4 Verschiedene Lernarten und -techniken

Wie bereits beschrieben soll beim maschinellen Lernen durch Computer neues Wissen durch die Benutzung von Eingabeinformationen konstruiert werden. Es gibt mehrere Ansätze, die dabei verfolgt werden können. Im Folgenden sollen nun vier Arten des maschinellen Lernens beschrieben werden:

- Klassifizierendes Lernen
- Assoziierendes Lernen
- Clustering
- und Numerische Vorhersage.

Das Verfahren beim klassifizierenden Lernen kennzeichnet sich dadurch, dass zunächst Beispiele als Eingabedaten verwendet werden, die bereits klassifiziert sind. Aus diesen soll der Computer lernen, unbekannte Beispiele zu klassifizieren. Hierbei geht es um die Vorhersage einer diskreten Klasse. Da dem Verfahren diese vorgegebenen Klassifikationen zum Trainieren zur Verfügung gestellt werden, wird klassifizierendes Lernen auch „überwachtes“ Lernen (supervised learning) genannt. Beim überwachten Lernen nimmt man an, dass ein „Lehrer“ den Lernvorgang überwacht, indem er bereits klassifizierte Beispiele, die als korrekt zugeordnet gelten, als Ausgangsbasis des Lernvorgangs zur Verfügung stellt und die Ergebnisse aufgrund seines Wissen als korrekt oder inkorrekt klassifiziert erkennen kann. Der Erfolg wird an neu klassifizierten Beispielen gemessen, bei denen die Klassen dem Algorithmus nicht bekannt sind. Dabei erfolgt die Bewertung so, dass die gelernte Konzeptbeschreibung für eine unabhängige Testdatenmenge ausprobiert wird, wobei die richtigen Klassifikationen im Voraus bekannt sind, aber der Maschine nicht zur Verfügung gestellt werden. Die Erfolgsrate für die Testdaten bietet objektives Maß dafür, wie gut das Konzept erlernt wurde. Die Fehler werden identifiziert, indem die Soll-Vorgaben mit der Systemausgabe durch den „Lehrer“ verglichen werden [vgl. WITTEN ET AL. 2001:42f]. Bestenfalls ergeben sich daraus Klassifikationsregeln, allerdings nur, wenn alle Auswertungen erfolgreich sind. Eine bekannte Vorgehensweise stellt die Kreuzvalidierung dar. Hierbei wird die Beispielmenge in *n* gleich große Teilmengen geteilt. Es folgen *n* Lernläufe, bei denen jeweils eine der Teilmengen die Testmenge bildet, die Vereinigung der übrigen bildet die Trainingsmenge. Verbreitet ist die so genannte 10-fache Kreuzvalidierung. Dabei gibt es zehn

Lernläufe, bei denen jeweils mit 9/10 der Beispiele gelernt wird und 1/10 als Testmenge verwendet wird. [vgl. WITTEN ET AL. 2001:128ff].

Beim assoziierenden Lernen geht es um das Erschließen von Assoziationen zwischen Merkmalen, nicht nur um die Vorhersage eines bestimmten Klassenwertes. Dabei sollen vielmehr „interessante“ und signifikante Strukturen gefunden als Vorhersagen getroffen werden, was, wie weiter oben bereits beschrieben, ebenfalls ein Ziel maschinellen Lernens darstellt. Im Vergleich zu Klassifikationsregeln können Assoziationsregeln die Werte für jedes einzelne Attribut „vorhersagen“. Es können dabei auch mehrere mögliche Attributkombinationen gelernt werden [vgl. WITTEN ET AL. 2001:68].

Anders als bei den vorherigen Lernarten werden beim Clustering Gruppen, sogenannte Cluster, von zusammengehörigen Beispielen gesucht. Es sollen Cluster gefunden und diesen dann Instanzen zugeordnet werden. Dabei werden ähnliche Instanzen, deren Klassenzugehörigkeit nicht bekannt ist, gruppiert. Anstelle von Klassifizierungen werden Cluster gelernt. Bei der Ausgabe wird dann gezeigt, wie die Instanzen den Cluster zugeordnet sind. Clustering ist „unüberwacht“ (unsupervised learning). Es gibt keinen „Lehrer“, der die Solldaten oder eine Bewertung liefert. Clustering wird zum Beispiel häufig in der Biologie eingesetzt. Clustering-Verfahren werden im Verlauf dieser Arbeit später noch genauer beschrieben.

Die numerische Vorhersage stellt eine Variante des klassifizierenden Lernens dar. Allerdings ist das Ergebnis keine diskrete Klasse, sondern ein numerischer Wert, vergleichbar mit numerischen Klassen. Das Lernen erfolgt überwacht, so dass eine Vorlage mit einem Zielwert bereits vorhanden ist. Der Erfolg des Lernvorgangs wird über die Testdaten ermittelt [vgl. WITTEN ET AL. 2001:75f]. Sollte keine Klasse für eine Instanz angegeben werden, werden die zusammengehörigen über Clustering-Verfahren gruppiert. WITTEN ET AL. (2001:42ff) beschreiben, dass oftmals der vorhergesagte Wert für neue Instanzen weniger im Blickpunkt steht, als die Struktur der gelernten Beschreibung. Fokus liegt also eher darauf, welche Attribute die wichtigsten sind und in welchem Verhältnis sie zu dem numerischen Ergebnis stehen. [vgl. WITTEN ET AL. 2001:42f].

Dies soll Grundlage zur weiteren Erläuterung von maschinellen Lernvorgängen von Ontologien bilden, die in den Folgenden Kapiteln behandelt werden.

4 Ontology Learning

Im folgenden Kapitel geht es um Ontology Learning, welches zu der Entwicklung von Ontologien für das Semantic Web beitragen kann, hier aber unter der Berücksichtigung der Verwendung für MyShelf betrachtet wird. Dabei wird zunächst erklärt, was Ontology Learning ist und welche Aufgaben dadurch abgedeckt werden sollen. Der Ablauf beim maschinellen Lernen von Ontologien soll anhand von Bestandteilen eines Lernsystems vorgestellt werden. Des Weiteren werden allgemeine Lernansätze erläutert, sowie spezifische Ansätze, die im weiteren Verlauf dieser Arbeit zur Durchführung von Versuchen in Betracht gezogen werden.

4.1 Was ist Ontology Learning?

"Computers have promised us a fountain of wisdom but delivered a flood of data."

zitiert nach FRAWLEY ET AL. (1992:57)

Wie in Kapitel 2 schon beschrieben, stützt sich das Semantic Web auf Ontologien, die zugrunde liegendes Wissen strukturieren, damit man dieses für Maschinen verständlich und lesbar einsetzen kann. Der Erfolg des Semantic Web ist somit stark an die Qualität von Ontologien gebunden, welche schnell und einfach zu erstellen sein sollen [vgl. MAEDCHE; STAAB 2001:72ff]. Im Internet bilden derzeit strukturiertes Wissen und Daten eher die Ausnahme, als die Regel, so dass eine strukturelle Aufbereitung der Daten im WWW durchgeführt werden muss [vgl MAEDCHE; STAAB 2001:73]. Da nicht nur eine implizite Perspektive auf das enthaltene Wissen ausreichend ist, müssen Wege gefunden werden, um Konzepte in einer Ontologie ausführlich zu beschreiben. Allerdings kann diese ausführliche Beschreibung zu Problemen führen. Aufgrund der Fülle an Daten führt eine manuelle Verarbeitung der Informationen zu einem so genannten „knowledge acquisition bottleneck“

(Engpass bei der Wissensakquisition). Dies bedeutet, dass ein menschlicher Benutzer nicht in der Lage ist, die stetig nachrückende Menge an Informationen mit Metainformationen zu versehen. Da die „Handarbeit" bei der Erstellung von Ontologien einen erheblichen Aufwand darstellt, versucht man, maschinelle Lerntechniken zur Wissensakquisition hinzuzuziehen, um diesen Prozess zu automatisieren. Ontology Learning soll den Entwickler bei der Erstellung von Ontologien unterstützen:

> *"Ontology learning […] is an emerging field aimed at assisting a knowledge engineer in ontology construction and semantic page annotation with the help of machine learning […] techniques."*
>
> [OMELAYENKO 2001]

Wie vorher beschrieben enthalten Ontologien Wissen über Konzepte einer Domäne. Wie der Name Ontology Learning bereits sagt, handelt es sich um maschinelles Lernen von Ontologien. Bei der Erstellung von Ontologien ist explizites Domänenwissen eine Voraussetzung, um eine repräsentative Ontologie für diese Domäne zu erhalten. Da die manuelle Erstellung von Ontologien immer noch einen sehr arbeits- und zeitaufwendigen Prozess darstellt, beschreiben MAEDCHE & STAAB (2004) eine Methode, die sich als hilfreich erwiesen hat, nämlich die Integration maschinellen Lernens in diesen Erstellungsprozess [vgl. MAEDCHE, STAAB 2004:173f]. Auch wenn einige Applikationen zum maschinellen Lernen gereift sind, liegt ein voll automatischer Erwerb von Ontologien in weiter Ferne. Demzufolge findet der Erstellungsprozess semi-automatisch mit menschlicher Interaktion statt. Diesem Ansatz liegt das Paradigma des „Balanced Cooperative Modeling" nach MORIK ET AL. (1993) zugrunde. Dies beschreibt die koordinierte Interaktion und den Austausch zwischen Mensch (Wissensingenieur) und Maschine (Lernalgorithmus) bei der Erstellung von Ontologien.

> *"If the user as well as the system can perform a task, construct knowledge items of a certain kind, run (learning) tools, and revise given knowledge, then we call such a system balanced cooperative".*
>
> [MORIK 1994:299]

Balanced Cooperative Modeling stellt dabei einen flexiblen Einsatz von Tools vor, die einen Benutzer insofern unterstützen sollen, als dass entweder dieser oder das System die Wissensbasis mit Informationen versorgen kann, da beide gleichwertig verschiedene Aufgaben und Operationen durchführen können [vgl. MORIK 1994:317]. Die Untersuchung

der Wissensbasis, das Aufdecken von Widersprüchen und die Verfeinerung von Regeln können sowohl von einem Benutzer, als auch von dem System vollzogen werden. Der Benutzer wählt den Zeitpunkt, wann das System eine Aufgabe durchführen soll und wann er Aufgaben selbst übernimmt. In beiden Fällen werden dieselben Wissensrepräsentationen und Operationen angewandt [vgl. MORIK 1994:319].

4.2 Aufgaben und Ablauf des Ontology Learning

Um maschinelles Lernen von Ontologien zu ermöglichen, müssen maschinelle Lerntechniken so verwendet werden, dass eine Ontologie automatisch erstellt werden kann. Allerdings bleibt anzumerken, dass die resultierende Ontologie die Eigenschaften einer manuell erstellten Ontologie bezüglich des enthaltenen Domänenwissens aufweisen sollte. OMELAYENKO (2001) beschreibt die Eigenschaften von Lernmethoden so, dass der Wissensingenieur derart einbezogen wird, dass dessen spezifisches Domänenwissen genutzt werden kann und dieser bei der Erstellung von Ontologien durch das Lernsystem unterstützt wird. Dies erfordert die Lesbarkeit und Einsicht in alle Resultate, sowohl intern, als auch extern. Mit Lesbarkeit ist gemeint, dass die Ergebnisse in eine für den Benutzer ebenso verständliche Formalisierung gebracht werden, wie für den Computer. Dabei sollen die (Zwischen-)Ergebnisse während des Erstellungsprozesses genauso zur Verfügung stehen wie die Endergebnisse [vgl. OMELAYENKO 2001].

OMELAYENKO (2001) beschreibt unterschiedliche Aufgaben, die durch Ontology Learning abgedeckt werden können. Er erläutert, dass es nach LOPEZ (1999) Richtlinien zur manuellen Erstellung von Ontologien gibt, die der Beschreibung der Vorgehensweise in Kapitel 2.3 ähnlich sind. Allerdings werden in der unter Kapitel 2.3 genannten Vorgehensweise die Anforderungen für maschinelle Verfahren nicht berücksichtigt. Da das Wissen letztendlich vom Experten und seiner Erfahrung abhängt, überträgt OMELAYENKO (2001) dessen Aufgaben auf den Einsatz von Methoden zum Ontology Learning. Im Folgenden werden Aufgabenbereiche gezeigt, bei denen eine maschinelle Unterstützung mit Hilfe von Lerntechniken bei der Erstellung von Ontologien hilfreich sein kann. Die ersten drei Aufgaben beziehen sich auf den Erwerb der Ontologie, die restlichen auf die Pflege und Instandhaltung der Ontologien:

- **Erstellung der Ontologie**: Die Erstellung einer Ontologie erfolgt von Anfang an durch einen Wissensingenieur, der dabei durch maschinelle Lerntechniken unterstützt wird, indem ihm Vorschläge zur Auswahl wichtiger Konzepte und Relationen gemacht werden.

- **Extraktion von Konzepten**: Bei der Extraktion von relevanten Konzepten aus Dokumenten sollen Lernsysteme Metadaten verarbeiten und daraus eine Ontologie generieren, was dann wahlweise auch mit Unterstützung des Wissensingenieur erfolgen kann.

- **Extraktion von Instanzen**: Ein weiterer Schritt stellt die Extraktion von Instanzen zu bestimmten Konzepten aus einer Dokumentenmenge dar. Hierbei werden bekannte Methoden aus dem Bereich Information Extraction verwendet. Dabei werden Dokumente mit Annotationen versehen, welche zur Beschreibung bestimmter Informationen innerhalb der Dokumente verwendet werden.

- **Integration der Ontologie**: Durch automatisches Lernen von Ontologien kann eine sehr große Wissensgrundlage entstehen. Aufgabe könnte es dann sein, die erstellte Ontologie in eine bestehende Wissensbasis zu integrieren.

- **Aktualisierung der Ontologie**: Zur Pflege und Aktualisierung der Ontologie können Lernverfahren eingesetzt werden, um Änderungen, die z.B. innerhalb der Instanzen (Webseiten) durchgeführt wurden und somit dann auch Auswirkungen auf Konzepte und Relationen der Ontologie haben können, automatisch zu erkennen und diese in der Ontologie anzugleichen.

- **Verfeinerung der Ontologie**: Diese stellt eine automatisierte Modifizierung kleinerer Relationen dar, welche die Ontologie selbst präziser machen soll. Dabei werden jedoch keine allgemeinen Konzepte oder Strukturen verändert. Im Gegensatz zur Aktualisierung geht es hier um Relationen, die bei einer Aktualisierung nicht berücksichtig werden, weil sie eventuell dafür nicht vorgesehen waren.

Der Ablauf des Prozesses beim Ontology Learning erfolgt analog zum Data Mining Prozess, der in Kapitel 3.1 vorgestellt wurde. Dieser soll im Folgenden Kapitel veranschaulicht werden.

4.3 Architektur eines semi-automatischen Ontology Learning Systems

Nachdem eine Vielzahl an Einsatzmöglichkeiten von maschinellen Lerntechniken für die Erstellung von Ontologien soeben vorgestellt wurden, sollen die Schritte des Lernprozesses anhand der Architektur eines Ontology Learning Systems beschrieben werden. Dies soll durch einen Ansatz von MAEDCHE & STAAB (2004) veranschaulicht werden, der mögliche Bestandteile und Grundkomponenten eines Systems zum Ontology Learning beschreibt [vgl. MAEDCHE; STAAB 2004:174]. Da diese Komponenten im Laufe des Lernprozesses spezielle Aufgabenbereiche abdecken, können anhand der Module und ihrem Zusammenspiel wesentliche Schritte des Lernprozesses erläutert werden.

Basierend auf den Phasen des Data Mining Zyklus baut die Architektur für ein Lernsystem darauf auf, dass Eingabedaten zunächst aufbereitet werden, um danach in einer passenden Form von einem Lernalgorithmus verarbeitet werden zu können. Die vorgestellte Architektur eines Lernsystems für Ontologien beinhaltet demnach vier wesentliche Komponenten (siehe Abbildung 6):

- Komponente zum Ontology Management
- Komponente zur Textverarbeitung (Resource Processing)
- Algorithmen-Bibliothek (Algorithm Library)
- Komponente zur Koordination (Coordination Component)

Mit Hilfe der Ontology-Management-Komponente können Ontologien manuell bearbeitet werden. Dabei können die Ontologien weiterentwickelt und Veränderungen vorgenommen werden. Von dieser Komponente aus können Eingabedaten (Textdokumente, Datenbanken, vorhandene Ontologien) vom Wissensingenieur ausgewählt werden. Auch erhält er über diese Komponente Zugriff auf Methoden zur Vor- und Aufbereitung von Texten, als auch auf Algorithmen der Algorithmen-Bibliothek [vgl. MAEDCHE; STAAB 2001:74].

Nachdem die Eingabe der Daten ermöglicht wurde, sollen diese bei dem Modul zum Resource Processing derart aufbereitet werden, dass die Daten für den Lernalgorithmus in einer verwertbaren Form vorliegen. Die Eingabedaten, die später zum Einsatz mit den Algorithmen kommen, werden in dieser Komponente aufbereitet. Es werden Techniken angewandt, welche die Inputdaten linguistisch analysieren und kennzeichnen. Dabei stellt die Hauptaufgabe die Generierung eines aufbereiteten Datensatzes dar, der in seiner aufbereiteten

Form von Algorithmen benutzt wird [vgl. MAEDCHE; STAAB 2004:174ff]. Ein wichtiger Bestandteil dieser Komponente ist die Verarbeitung natürlicher Sprache (Natural Language Processing oder abgekürzt NLP). Um den extrahierten Text anschließend weiterzuverarbeiten, werden je nach Sprache der Eingangsdaten NLP-Systeme verwendet. Da die semantischen Zusammenhänge von Wörtern in Sätzen für den späteren Lernversuch wichtig sind, werden die Texte annotiert. Hierbei versehen NLP-Systeme die Eingabedaten mit einer Annotation, so dass einzelne lexikalische Bestandteile des Textes durch Tags als solche beschrieben werden. Dieser Schritt erleichtert es, Algorithmen auf spezielle Satzteile oder linguistische Einheiten zugreifen zu lassen. Nach der Vorverarbeitung des Textes bringt das Resource Processing Modul die Eingabedaten in eine Form, die spezifisch für die Weiterverarbeitung durch den jeweiligen Algorithmus ist [vgl. MAEDCHE; STAAB 2004:177].

Ein weiterer Bestandteil der Architektur ist die Algorithmen-Bibliothek. Sie beinhaltet Algorithmen für das maschinelle Lernen von Ontologien. Bei der Anwendung verschiedener Algorithmen werden aufbereitete Daten als Input zur Erstellung einer Ontologie verwendet. Das Ergebnis wird in einer standardisierten Form ausgegeben, so dass verschiedene Resultate miteinander kombiniert werden können [vgl. MAEDCHE; STAAB 2004:175].

Die Coordination Component ermöglicht die Koordination der anderen Komponenten. Der Wissensingenieur interagiert über sie mit der Lernkomponente. Sie stellt eine grafische Benutzeroberfläche zur Verfügung, über die der Benutzer die Eingabedaten, die im weiteren Verlauf verarbeitet werden sollen, auswählt. Auch dient sie der Hilfe bei der Anwendung von Vorverarbeitungsmethoden von Texten.

Es werden intuitiv verständliche Benutzerschnittstellen angeboten, die bei der Auswahl der relevanten Daten helfen, die Methoden zur Aufbereitung von Texten anwenden oder einen spezifischen Extraktionsmechanismus starten. Auch der Start der Mechanismen der Extraktionsprozedur kann von hier aus gestartet werden. All dies vereint diese Komponente. Folglich werden die Ergebnisse hier auch zusammengeführt und dem Benutzer präsentiert [vgl. MAEDCHE; STAAB 2004:175].

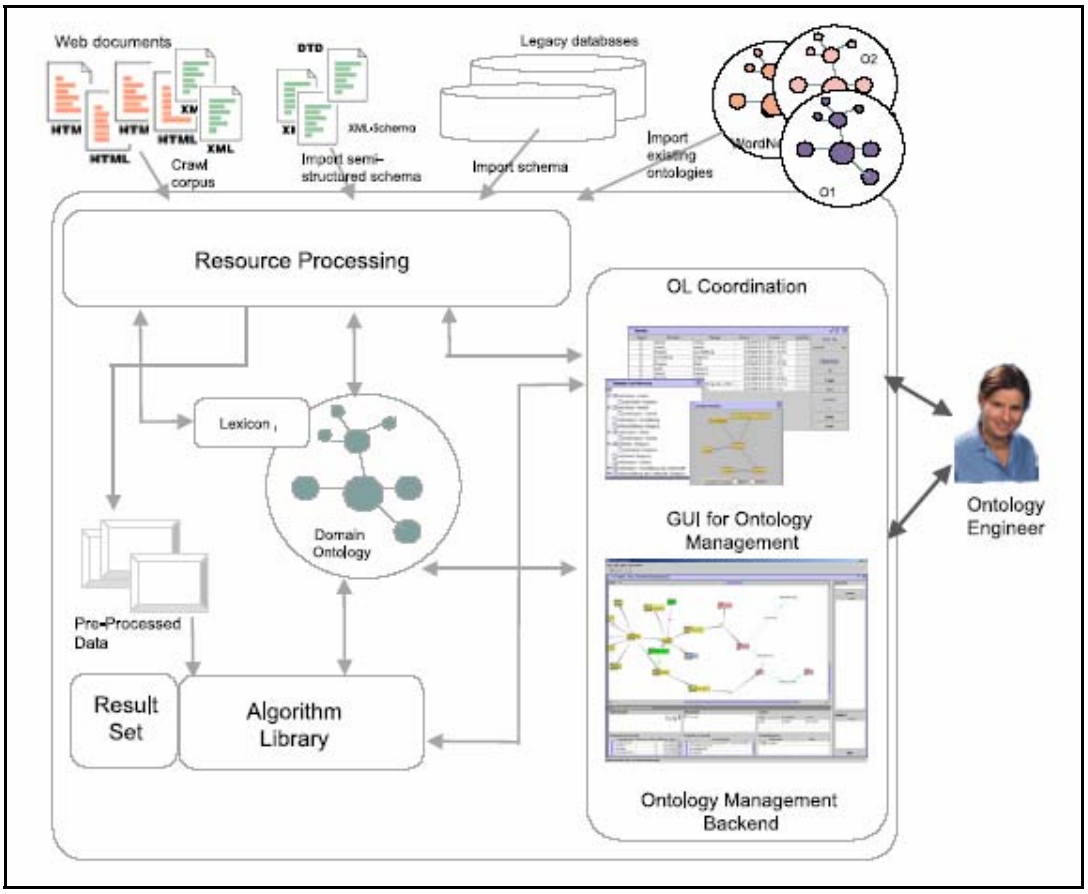


Abbildung 6: Architektur eines Lernsystems nach MAEDCHE & STAAB (2004:175).

Der Unterschied zur Ontology Management Komponente liegt darin, dass hier der allgemeine Zugriff und die Benutzeroberfläche zur Verfügung stehen. Die Ontology-Management-Komponente ist für die Darstellung der Ontologie und ihrer Bestandteile zuständig. Die Komponenten bilden gemeinsam ein System, das alle wichtige Schritte zum Ontology Learning vereint und die Ergebnisse zugänglich macht.

Wie man sieht, liegt dem Architekturansatz das Prinzip des Data Mining Prozesses – von der Vorverarbeitung der Daten über die Anwendung von Lernalgorithmen, bis hin zur Präsentation des Ergebnisses – zugrunde. Diese Systemarchitektur stellt den prinzipiellen Aufbau eines Lernsystems vor. Dabei gilt es zu beachten, dass die beschriebene Architektur unabhängig von den jeweiligen Lernalgorithmen ist. Im Folgenden werden maschinelle Lernansätze zum Einsatz für das Ontology Learning vorgestellt.

4.4 Allgemeine Ansätze zum Ontology Learning

Im Folgenden werden einige Lernansätze für das Ontology Learning beschrieben. Dabei geht es darum, relevante Konzepte einer Domäne zu identifizieren und hierarchisch in Form einer Taxonomie anzuordnen, um auf diese Weise die Relationen zwischen den Konzepten zu beschreiben. MAEDCHE & STAAB (2004) stellen diverse Algorithmen zum maschinellen Lernen von Ontologien vor.

4.4.1 Extraktion von Konzepten

Um eine Ontologie maschinell erstellen zu lassen, müssen aus einer Datenmenge Wörter und Begriffe gefunden werden, die sich für die Bildung einer Taxonomie als Konzepte eignen. Eine Technik zur Extraktion relevanter lexikalischer Einträge, die Hinweise auf relevante Konzepte für die Ontologie beinhalten könnten, beschreiben MAEDCHE & STAAB (2004:178):

> *"In general this approach is based on the assumption that a frequent term in a set of domain-specific texts indicates occurrence of a relevant concept."*

Somit sollen häufig vorkommende Terme anhand der Termfrequenz in einem annotierten Korpus untersucht werden. Da die reine Termfrequenz alleine nicht aussagekräftig genug ist, wird zur Identifizierung charakteristischer Wörter eine Gewichtung dieser Wörter benötigt. Hierfür wird das Standardmaß tfidf (Termfrequenz/Inverse Dokumentfrequenz) verwendet. Die Termfrequenz gibt die Häufigkeit eines Terms in einem Dokument an. Dabei haben häufig vorkommende Wörter eine wichtigere Bedeutung als Wörter, welche seltener enthalten sind. Bei der inversen Dokumentfrequenz wird Wörtern, die nur in wenigen Dokumenten vorkommen innerhalb des Korpus eine größere Bedeutung zugemessen als selten vorkommenden Wörtern.

- Die Termfrequenz $\mathbf{lef}_{\mathbf{l}\ \mathbf{d}}$ wird dadurch ermittelt, wie häufig dieses Wort **l** in einem Dokument vorkommt **d**.
- Die Dokumentfrequenz $\mathbf{df}_{\mathbf{l}}$ ist die Anzahl an Dokumenten im Korpus **D**, in welchen das Wort **l** vorkommt.
- Die Korpusfrequenz $\mathbf{cf}_{\mathbf{l}}$ ist die Gesamtzahl des Vorkommens von **l** im gesamten Korpus **D**.

Somit ergibt sich für das Maß tfidf:

$$\mathbf{tfidf_{l,d} = lef_{l,d} * log} \left(\frac{|D|}{df_l} \right)$$

Für die Ermittlung von tfidf für einen gesamten Korpus werden die einzelnen Werte für tfidf addiert. Bei der Gewichtung wird somit berücksichtigt, dass Terme, die zu häufig oder zu selten in fast allen Dokumenten vorkommen, an Gewicht verlieren, als Terme, die über ein gesamtes Korpus gleich verteilt vorkommen [vgl. MAEDCHE; STAAB 2004:178ff].

4.4.2 Extraktion taxonomischer Relationen

Hierbei geht es darum, Relationen zwischen Termen zu ermitteln, die in einer hierarchischen Beziehung zueinander stehen. Dies kann auf unterschiedliche Arten erfolgen. In dem hier genannten Ansatz werden hauptsächlich die folgenden drei Techniken verwendet [vgl. MAEDCHE; STAAB 2004:178ff]:

- eine auf Statistik basierende Extraktion von Konzepten mit Hilfe von Clustering,
- eine auf Statistik basierende Extraktion von Konzepten mit Hilfe von Klassifizierung,
- und die Verwendung lexiko-syntaktischer Muster zur Extraktion von Konzepten.

Clustering

Clustering-Algorithmen gruppieren Konzepte aufgrund der Bewertung ihrer Ähnlichkeit oder Unterschiedlichkeit zueinander. Durch das schrittweise Vorgehen versucht der Algorithmus eine hierarchische Struktur derjenigen Konzepte zu erstellen, die bisher noch nicht in der Hierarchie berücksichtigt wurden. Das Ziel besteht darin, die Objekte in eine Hierarchie von Klassen einzuordnen und dabei die Ähnlichkeiten von gruppierten Konzepten zu maximieren, die einer bestimmten Kategorie angehören. [vgl. OMELAYENKO 2001]. Dabei werden so genannte Cluster erstellt, die Elemente, welche ähnlich sind, gruppiert. Kriterien, die hierbei zur Überprüfung der Ähnlichkeit von Konzepten herangezogen werden, sind Daten über das Vorkommen und die Häufigkeit von Wörtern in einem Korpus. Hierbei wird die Distribution nach HARRIS (1968) berücksichtigt. Diese wird in LAVELLI ET AL. (2004:615) beschrieben. Die Grundannahme ist, dass „[…] the semantics of a term is conveyed by the terms that co-

occur with it (i.e. that occur in the same documents). The basic intuition behind this representation is the so-called distributional hypothesis, formulated by the well-known linguist Zellig Harris, which states that terms with similar distributional patterns tend to have the same meaning." Somit werden beim Clustering Terme berücksichtigt, die im Kontext anderer Terme auftreten, weil man davon ausgeht, dass diese dazu tendieren, die gleiche Bedeutung zu haben. Somit werden beim Clustering Objekte gruppiert, deren Mitglieder bezüglich dieser Distribution semantisch ähnlich zueinander sind. Allgemein soll größtmögliche Homogenität innerhalb der Cluster und größtmögliche Heterogenität zwischen diesen bestehen. Mit jedem dieser Verfahren kann man eine Hierarchie von Termgruppierungen erstellen, die dann vom Benutzer genau geprüft werden können.

Klassifizierung

Klassifizierung kann dazu eingesetzt werden, um eine bestehende domänen-spezifische Taxonomie zu verfeinern. Bei der Klassifizierung geht es um die Vorhersage einer diskreten Konzeptklasse. Man benötigt eine bereits bestehende umfangreiche Hierarchie und klassifiziert neue relevante Terme in die gegebene Konzepthierarchie. Die bereits beschriebene Distribution von Termen nach HARRIS (1968) wird dazu benutzt, den Klassifizierer aus einem Trainingskorpus und den vordefinierten, bereits existierenden Konzepten lernen zu lassen. Der so trainierte Algorithmus kann dann auf eine Testmenge angewendet werden, in der sich nicht-klassifizierte Terme befinden. Somit soll der Klassifizierer Vorschläge liefern, welche neuen Terme zu welchen Konzepten gehören. Typische Lernalgorithmen zur Klassifizierung sind K Nearest Neighbor (kNN) und Support Vector Machines, deren genaue Beschreibung z.B. in WITTEN ET AL. (2001) gefunden werden kann. [vgl. MAEDCHE; STAAB 2004:180].

Lexiko-Syntaktische Muster

Zur Extraktion taxonomischer Beziehungen zwischen Konzepten wird der Ansatz von MARTI A. HEARST (1992) vorgeschlagen. Dabei wird in einem Korpus nach lexikalischen und syntaktischen Mustern gesucht und diese Muster in Form von regulären Ausdrücken definiert. Reguläre Ausdrücke sind Muster, mit denen man den Aufbau von Zeichenketten beschreiben kann. Ein regulärer Ausdruck ist also eine Beschreibung, die auf eine ganze Klasse von Zeichenketten zutrifft. MAEDCHE & STAAB (2004:180) definieren den Ansatz wie folgt:

> *"Define a regular expression that captures re-occurring expressions and map the results of the matching expression to a semantic structure, such as taxonomic relations between concepts."*

Es soll also ein regulärer Ausdruck definiert werden, der wiederholt enthaltene Muster aus einer Dokumentenmenge erfasst. Bei Übereinstimmungen des regulären Ausdrucks mit Mustern wird das Ergebnis übereinstimmender Ausdrücke, also Konzepte, in einer hierarchischen Struktur abgebildet [vgl. MAEDCHE; STAAB 2004:180]. Das Korpus wird nach Instanzen unterschiedlicher lexiko-syntaktischer Muster durchsucht, um is-a-Beziehungen zwischen Konzepten zu extrahieren.

4.4.3 Extraktion binärer Beziehungen

Extraktion allgemeiner binärer Beziehungen bedeutet, dass Konzepte gesucht werden, die wechselseitig in Beziehung stehen. Um die Beziehungen von relevanten Konzepten in einem Korpus zueinander herauszufinden, können hierfür Assoziationsregeln Anwendung finden. Mit Assoziationsregeln können unterschiedliche Kombinationen von Konzepten ausgedrückt werden [vgl. WITTEN ET AL. 2001:68]. Dabei wird untersucht, inwieweit ein Konzept von anderen Konzepten abhängig ist bzw. in welcher Relation ein Konzept zu den anderen steht. Assoziationsregeln zielen darauf ab, interessante Beziehungen und Verbindungen zwischen einzelnen Bestandteilen einer großen Datenmenge zu finden. Für das maschinelle Lernen von Ontologien können Assoziationsregeln auf syntaktische Strukturen und deren Vorkommen in einem Korpus angewendet werden [vgl. MAEDCHE, STAAB 2004:181]. Dabei sollen Begriffspaare extrahiert werden, bei denen linguistische Abhängigkeiten voneinander ausgewertet werden. Es kann zusätzliches Wissen aus einer bereits bestehenden Begriffshierarchie berücksichtigt werden. Anschließend sollen die häufigsten Begriffsmengen berechnet werden. Auf der Basis der Ergebnisse werden dann Assoziationen für die Begriffe abgeleitet [vgl. STAAB ET AL. 2003]. STAAB ET AL. (2003) stellen dies am folgenden Beispiel vor:

Mecklenburg's schönstes Hotel liegt in Rostock.

Die *Region* Mecklenburg wird also mit dem Konzept *Hotel* in Beziehung gestellt, Hotel wiederum mit dem Konzept *Stadt* (Rostock), so dass sich die Beziehungen (Region, Hotel) und (Hotel, Stadt) ergeben. Wenn nun Hintergrundwissen aus bestehender Ontologie einer

Tourismusdomäne hinzugezogen wird, so kann man zum Beispiel daraus entnehmen, dass *Hotel* ein Unterkonzept zu *Unterkunft* darstellt, und *Unterkunft* wiederum ein Unterkonzept von *Organisation* ist. Somit werden für das Begriffspaar (Region, Hotel) auch die Paare (Region, Unterkunft) und (Region, Organisation) untersucht. Analog erfolgt dies bei den Konzepten (Hotel, Stadt). Stadt ist eine Unterkategorie von Gebiet. Somit wird hierfür aus der Tourismusontologie die Beziehung (Hotel, Gebiet) untersucht. Auf Basis der häufigsten Begriffspaare werden dann die Begriffsassoziationen abgeleitet.

4.4.4 Pruning

Pruning bezeichnet eine Vorgehensweise, bei der hinsichtlich einer Ontologie bestimmte Konzepte „herausgeschnitten" werden, weil man bei diesen davon ausgeht, dass sie für die Ontologie nicht relevant sind. Das Pruning von Ontologien wird relevant, wenn man auf bereits bestehenden Ontologien aufbaut (z.B. lexikalisch-semantische Ontologien wie WordNet) [vgl. STAAB ET AL. 2003]. Dabei wird angenommen, dass die Frequenz spezifischer Konzepte und Relationen in Dokumenten für die Entscheidung sehr wichtig ist, ob ein gegebenes Konzept oder eine Relation in der Ontologie erhalten bleiben soll, oder nicht. Es wird ein Ansatz angewandt, bei welchem die Frequenzen der Konzepte und deren Relationen bestimmt werden. Begriffe, die in einem Korpus häufig vertreten sind, werden folglich als Bestandteil einer Domäne angesehen und als relevant erachtet. Allerdings reichen die reinen Häufigkeitswerte nicht aus, um eine Aussage über die Wichtigkeit und den Bezug der Begriffe zueinander darzustellen. Deswegen werden Konzepte aus einer anderen Domänenontologie hinzugezogen und mit den Konzeptfrequenzen aus dem verwendeten Korpus verglichen. Der Pruning-Algorithmus verwendet hierbei die berechneten Frequenzen und Maße, um relevante Konzepte zu bestimmen. Alle existierenden Konzepte und Relationen, die häufiger in dem domänen-spezifischen Korpus vorkommen, bleiben in der Ontologie. Der Benutzer kann also das Pruning auf diejenigen Konzepte anwenden, die weder im bestehenden domänen-spezifischen noch im allgemeinen Korpus enthalten sind [vgl. MAEDCHE; STAAB 2004:183].

4.5 Lernansätze im Kontext dieser Arbeit

Es sollen nun im Weiteren Ansätze im Detail präsentiert werden, die später im Zusammenhang mit maschinellem Lernen von Ontologien für MyShelf verwendet oder in Betracht gezogen werden. Darunter fallen ein Ansatz, der auf Basis heterogener Quellen arbeitet, die formale Beschreibungsanalyse, sowie die Verwendung von Mapping-Regeln zur Erstellung einer Ontologie.

4.5.1 Lernen aus heterogenen Beweisquellen

CIMIANO ET AL. (2004) stellen einen Ansatz vor, der das „knowledge acquisition bottleneck“ zumindest teilweise überwinden soll, indem Wissen unter Berücksichtigung verschiedener Beweisquellen für eine Ontologie automatisch erworben wird. Mit Beweisquelle ist gemeint, dass in diesen domänen-spezifischen Quellen Wissen über semantisch korrekte Konzeptrelationen bezüglich der spezifischen Domäne enthalten ist (hier: is-a-Beziehungen). Dieses Wissen ist letztlich für die Wissensakquisition in Ontologien entscheidend. Um nun Konzepte für Taxonomien aus unstrukturierten Beweisquellen (Textkorpora, Internetquellen) zu lernen, die eine is-a-Relation repräsentieren, werden Informationen benötigt. Bei diesem Ansatz werden die Informationen aus Hearst-Pattern, die in einem großen Textkorpus [vgl. HEARST 1992] oder im WWW in Verbindung mit WordNet gefunden werden, bezogen. Auch wird eine Heuristik „vertikaler Relationen“ verwendet. Diese Ansätze werden im Folgenden näher beschrieben.

Ein Beispiel soll zeigen, wie zwei Terme (*conference* und *event*) bezüglich der is-a-Relation taxonomisch auf verschiedene Arten in Beziehung zueinander stehen können:

- is-a (conference, event)
- is-a (event, conference)
- siblings(conference, event)

Allerdings ist es auch möglich, dass Terme nicht zwingend in Relation zueinander stehen müssen. Die is-a Beziehung drückt im ersten Beispiel aus, dass *conference* eine Unterkategorie von *event* darstellt, im zweiten Beispiel genau umgekehrt. Siblings bedeutet, dass beide Konzepte auf einer Ebene stehen und hierarchisch gleichgestellt sind. Allgemein

wird durch eine is-a-Beziehung beschrieben, dass ein Konzept eine Spezialisierung eines anderen Konzeptes ist. Wenn man erreicht, alle Terme in eine der oben beschriebenen und korrekten Beziehung zueinander zu stellen, erhält man eine korrekte Konzepthierarchie [vgl. CIMIANO ET AL. 2004]. In diesem Ansatz liegt der Fokus auf einfachen Klassifizierungen, d.h. es soll entschieden werden, ob zwei Terme in einer is-a-Beziehung stehen, oder nicht. Infolgedessen sollen so viele Beweisquellen wie möglich herangezogen werden und diejenigen Relationen für die Konzepthierarchie gewählt werden, die den deutlichsten Beweis bezüglich all der verschiedenen Bezugsquellen aufweisen können, nämlich dass bezüglich der Domäne betrachtet semantisch in Beziehung stehen [vgl. CIMIANO ET AL. 2004].

Im Folgenden werden die Beweisquellen und die Vorgehensweisen beschrieben, die bei diesem Ansatz benutzt werden.

4.5.1.1 Hearst-Pattern aus der Textkollektion

Für die Untersuchung verschiedener „Beweise“ werden lexiko-syntaktische Muster (siehe 4.4.2) betrachtet, die in einem Korpus enthalten sind. Diese Muster werden über reguläre Ausdrücke dargestellt und auf das Korpus angewendet. Semantische Relationen werden extrahiert, indem das Korpus nach verschiedenen dieser Muster, die als Relation für die Hierarchie in Frage kommen würden, durchsucht wird. Die Ergebnisse der übereinstimmenden Ausdrücke werden dann in einer Taxonomie als Relationen zwischen Konzepten dargestellt [vgl. MAEDCHE ET AL. 2003:302ff]. Der auf HEARST (1992) basierende Ansatz sieht folgendermaßen aus: es werden alle im Korpus vorkommenden Nominalphrasen mit nur einer Vergleichsphrase in eine is-a-Beziehung gestellt. Eine Nominalphrase ist eine syntaktische Kategorie, die in einem Satz entweder Subjekt- oder Objektfunktion ausfüllt. Diese Kombination zweier Terme als Konzeptkandidaten stellt ein Hearst-Pattern dar. Dabei wird die Häufigkeit dieses Relationspaares gezählt. Je höher die Frequenz eines Paares ist, desto relevanter ist diese Relation für die Hierarchie. Dieser Ansatz ähnelt der Vorgehensweise bei der unter 4.4.3 beschriebenen Extraktion binärer Relationen.

Bei all diesen Möglichkeiten wird für je ein Termpaar untersucht, wie oft dieses Hearst-Muster in dieser Kombination im Korpus vorkommt. Um diese Patterns zu finden, werden reguläre Ausdrücke mit Hilfe von POS-Tagging (Part-of-Speech), welches einzelne

Bestandteile in Sätzen identifiziert und mit Tags kennzeichnet, angewendet. Eine Reihe an Hearst-Pattern für den Begriff *conference* könnte dann folgendermaßen aussehen:

- is-a_{HEARST}(conference, event)
- is-a_{HEARST}(conference, body)
- is-a_{HEARST}(conference, meeting)
- is-a_{HEARST}(conference, course)
- is-a_{HEARST}(conference, activity)

Bei ihrem Versuch stellten CIMIANO ET AL. (2004) fest, dass trotz eines großen Korpus nur wenige Hearst-Pattern vorkamen, was für die Auswahl an relevanten Konzepten aus dem gesamten Korpus positiv ist. Dies bedeutet, dass zwar wenige Konzeptpaare relevant sind, diese wenigen jedoch aus domänen-spezifischer Sicht in die Hierarchie passen [vgl. CIMIANO ET AL. 2004].

4.5.1.2 Hearst-Pattern aus WordNet

Eine weitere Beweisquelle stellen die Informationen über Hyponymie-Beziehungen aus WordNet dar. WordNet ist eine englischsprachige lexikalische Datenbank, die semantische Beziehungen zwischen Wörtern beschreibt, und wurde am Cognitive Science Laboratory der Universität Princeton entwickelt:

> *"WordNet is an online lexical database designed for use under program control. English nouns, verbs, adjectives, and adverbs are organized into sets of synonyms, each representing a lexicalized concept. Semantic relations link the synonym sets."*
>
> [MILLER 1995:39]

Hyponymie bezeichnet die Beziehung eines untergeordneten Begriffes zu einem übergeordneten, z.B. verhält sich „*Blume*" zu „*Pflanze*" hyponym. Das Gegenstück hierzu ist Hyperonymie, wodurch die Beziehung eines übergeordneten Begriffes zu einem untergeordneten Begriff ausgedrückt wird, d.h. in dem Beispiel wäre dann „*Pflanze*" zu „*Blume*" hyperonym. Die Einbeziehung der semantischen Hyponymie-Beziehungen aus WordNet hat den Vorteil, dass dadurch automatisch eine hierarchische Struktur festgelegt

wird: „Because there is usually only one hypernym, this semantic relation organizes the meanings of nouns into a hierarchical structure" [vgl. MILLER 1995:40].

Zur Extraktion von is-a-Beziehungen in einem Korpus sollen bei diesem Ansatz Hearst-Pattern auf eine Hyponymie-Beziehung unter Berücksichtigung von WordNet untersucht werden. Da Wörter mehrere Bedeutungen haben können, können hier einmal alle Bedeutungen eines Terms zur Paarbildung und ein zweites Mal nur die erste und somit häufigste Bedeutung in WordNet herangezogen werden. Dabei muss beachtet werden, dass sich Wörter aufgrund von Synonymie zu mehreren Begriffen hyponym verhalten können, also Unterbegriffe von mehreren unterschiedlichen Oberbegriffen sein können. Dies kann auch die Beziehung zweier zu vergleichender Terme beeinflussen, wenn die Begriffe in einem anderen Hearst-Pattern vorkommen, aber darin komplett andere Oberbegriffe und semantische Bedeutungen haben. Im Gegensatz zu den anderen Beweisquellen stellt WordNet eigentlich keine unstrukturierte Quelle dar, da die enthaltenen Informationen domänen-unabhängig und deswegen auch sehr allgemein sind. Für den Ansatz wird dies allerdings vernachlässigt, weil WordNet bei einer spezifischen Domäne zum Einsatz kommt [vgl. CIMIANO ET AL. 2004].

Für is-a Beziehungen, die mit dieser Methode gelernt werden sollen ergibt sich, dass die Verwendung von WordNet semantisch bessere Ergebnisse liefert, wenn man sich auf die erste Bedeutung der verwendeten Terme für die Bildung von Hearst-Pattern beschränkt, weil somit Mehrbedeutungen vernachlässigt werden.

4.5.1.3 Heuristik „vertikaler Relationen"

Um beim Ansatz von CIMIANO ET AL. (2004) weitere is-a-Beziehungen ermitteln zu können, wird eine Heuristik nach einem Ansatz von VELARDI ET AL. (2001) eingesetzt, die im weiteren Verlauf Heuristik „vertikaler Relationen" genannt wird. VELARDI ET AL. (2001:271) beschreiben, dass Konzepte über semantische Beziehungen verbunden sind, die vertikal oder horizontal sein können: "Vertical relations [..] [gathers] the more general concepts; Horizontal relations […] [gathers] the similar concepts" [VELARDI ET AL. 2001:271]. Vertikale Relationen beziehen sich also auf allgemeinere, unspezifischere Konzepte. Im Grunde geht es darum, dass man zwei Terme hat, die man auf eine is-a-Beziehung unter Benutzung von Hearst-Pattern untersuchen will. Dabei werden allgemeinere Beziehungen der Terme untereinander berücksichtigt, z.B. *credit card* als Verallgemeinerung von *card*. Dadurch

können Konzepte in einer Hierarchie sehr allgemein angeordnet werden. Auch werden PartOf-Beziehungen betrachtet, die angeben, dass ein Begriff aus semantischer Sicht zu einem Konzept gehört. Dies ermöglicht die Einbeziehung von Mehrworttermen. Durch die Berücksichtigung allgemeinerer Konzepte und PartOf-Beziehungen werden diese generelleren vertikalen Relationen extrahiert. Abbildung 7 zeigt das Konzept *card* als syntaktischen Oberbegriff mehrerer Mehrwortterme zu *card*.

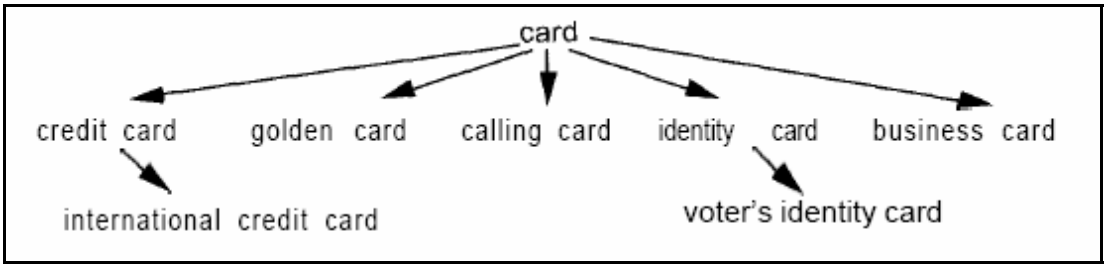


Abbildung 7: Vertikale Relationen mit Mehrworttermen in VELARDI ET AL. (2001:277).

Man erhält eine is-a-Beziehung dadurch, dass zu dem ersten Term ein zweiter in Bezug gestellt, der erste jedoch mit Adjektiven oder anderen beschreibenden Termen modifiziert wird, so dass dadurch Mehrwortterme herangezogen werden. Beispielsweise kann man "*international conference*" und "*conference*" in der Beziehung is-$a_{HEURISTIC}$(international conference, conference) darstellen. Somit werden aus semantischer Sicht allgemeinere Fassungen von Konzepten zusammengefasst und deren Mehrwortterme mit einbezogen. Die Zusammenhänge dieser Beziehungen werden dann über eine Heuristik realisiert [vgl. CIMIANO ET AL. 2004].

4.5.1.4 Hearst-Pattern aus Quellen des World Wide Web

Eine Erweiterung der Verwendung von Hearst-Patterns stellt das WWW dar. Problematisch bei der Arbeit mit Korpora sind selten vorkommende Terme, da diese domänen-spezifisch relevant sein könnten, es jedoch aufgrund ihrer niedrigen Frequenz bei der Hearst-Pattern-Bildung nicht in die Ontologie schaffen. CIMIANO ET AL. (2004) beschreiben jedoch, dass das Internet sich anbietet, dieses Problem zu beheben. Dafür wird ein Zugriff auf die Websuche von Google über Google API implementiert. Diese wird verwendet, um die Anzahl an Übereinstimmungen bestimmter Muster von Konzeptpaaren aus einem Korpus im Web zu ermitteln:

> *"With the Google Web APIs service, software developers can query more than 8 billion web pages directly from their own computer programs. The Google Web APIs service gives you query access to Google's web search, enabling you to develop software that accesses billions of web documents that are constantly refreshed.*
>
> [vgl. GOOGLE 2004]

An dieser Stelle soll Google API genutzt werden, um verschiedene Hearst-Pattern im Internet zu finden und die Anzahl der Hits und somit der Frequenz zu ermitteln. Die Hearst-Patterns stellen dabei die Indikatoren für eine gemeinsame taxonomische Relation dar. Dieser Ansatz ist der ursprünglichen Methode der Hearst-Patterns ähnlich, Hauptunterschied ist jedoch, dass bei den Hearst-Patterns je ein Muster auf ein Korpus angewendet wird. Bei diesem Ansatz wird jedoch für eine Kombination von Termen (t1, t2) eine gewisse Anzahl an Mustern pro Termpaar gesucht, die unterschiedliche Relationen der Terme zueinander darstellen. Diese werden als Suchanfragen an die Google API gesendet. Derartige Relationen können zum Beispiel so aussehen:

- <t1>s such as <t2>
- such <t1>s as <t2>
- <t1>s, including<t2>
- <t1>s, especially<t2>
- <t1> and other <t2
- <t1>s or other <t2>.

Mit diesen Anfragen werden die Vorkommnisse dieser Relationen für jeweils ein bestimmtes Termpaar im WWW untersucht. Für jedes Muster wird die Häufigkeit gezählt und anschließend verglichen. Als Ergebnis erhält man Termpaare, die in einer is-a-Beziehung zueinander stehen. CIMIANO ET AL. (2004) stellten fest, dass nur wenige Termkombinationen relevant sind. Aufgrund der enormen Größe des WWW und der unspezifischen Art seiner enthaltenen Quellen stellt das WWW eine sehr große allgemeine Quelle dar und ist nicht auf eine spezifische Domäne beschränkt. So traten bei diesem Ansatz sehr viele Fehler bezüglich der Übereinstimmung verschiedener Termpaare auf, da viele Terme in der gesuchten Kombination in keinerlei Beziehung zueinander standen [vgl. CIMIANO ET AL. 2004].

Trotz der Möglichkeit über die Muster Beziehungen zu finden, ist eine manuelle Überprüfung der Ergebnisse nötig, da man sich nicht zwangsläufig auf die is-a-Beziehungen verlassen kann [vgl. CIMIANO ET AL. 2004]. Obwohl die Strategie, die verschiedenen Lernansätze aus heterogenen Beweisquellen zu vereinen, sehr einfach gehalten ist, stellten CIMIANO ET AL. (2004) fest, dass sie zu einem besseren Ergebnis führen. Dies zeigt, dass es offenbar großes Potential bei der Fusion verschiedener Ansätze zum Lernen von is-a-Beziehungen gibt. Hauptherausforderung ist hier sicherlich die optimale Kombination zu finden.

4.5.2 Formale Begriffsanalyse

CIMIANO ET AL. (2003) stellen einen weiteren Ansatz für den automatischen Erwerb von Taxonomien oder Konzepthierarchien aus domänen-spezifischen Texten vor, welcher auf der formalen Beschreibungsanalyse (engl.: Formal Concept Analysis) basiert. Die formale Beschreibungsanalyse basiert ursprünglich auf GANTER & WILLE (1999). Ausgehend von einem Datensatz kann mit Hilfe der Beschreibungsanalyse eine Begriffshierarchie abgeleitet und visualisiert werden [vgl. STAAB ET AL. 2003a].

Ein weiterer Aspekt bei der Berücksichtigung von Merkmalen, der für die Extraktion von Konzepten zur Erstellung einer Hierarchie wichtig ist, sind die semantischen Zusammenhänge von Verben und deren Objekten. Man nutzt hierfür semantische Beschränkungen aus, die festlegen, welches Objekt zu einem Verb erlaubt ist. Diese spielen bei der Begriffsklärung von syntaktischen und lexikalischen Begriffen eine wichtige Rolle und werden auch Selektionsbeschränkungen genannt. In dem Satz „Das Kind sitzt auf der Bank“ wird beispielsweise durch die Selektionsbeschränkung des Wortes *sitzen* Bank als Sitzmöglichkeit und nicht als Geldinstitut erkannt [vgl. KORTMANN 1999:161f]. Man geht also davon aus, dass ein Verb starke Selektionsbeschränkungen voraussetzt, so dass eine Konzepthierarchie dadurch abgeleitet werden kann, dass die Beziehungen von Verben zu ihren Objekten aufgrund dieser Selektionsbeschränkungen bestimmt werden. Die Verben bilden dann die Konzepte, denen die Objekte untergeordnet werden. Da sie keine Nomen sind, werden diese Konzepte durch die Verben selbst mit der Erweiterung „able“ benannt, also zum Beispiel alle Objekte des Verbs „rent“ zu einer Klasse „rentable“ mit dem Objekt Apartment und der Unterkategorie „driveable“ zusammengefasst. Zum Beispiel existiert in einer Tourismusdomäne Wissen, dass bestimmte Begriffe nur mit bestimmten Aktivitäten in Bezug

gebracht werden können, wie z.B. „ein Auto kann gefahren werden, ein Apartment nicht“. Die folgende Matrix soll eine derartige Kombination beispielhaft verdeutlichen (siehe Tabelle 1).

	bookable	readable	driveable	rideable	joinable
apartment	**x**	**x**			
car	**x**	**x**	**x**		
motor-bike	**x**	**x**	**x**	**x**	
excursion	**x**				**x**
trip	**x**				**x**

Tabelle 1: Wissensmatrix aus dem Bereich Tourismus in CIMIANO ET AL. (2003:11)

Auf Basis dieses Wissens könnte man sehr einfach eine Konzepthierarchie erstellen wie in Abbildung 8 dargestellt.

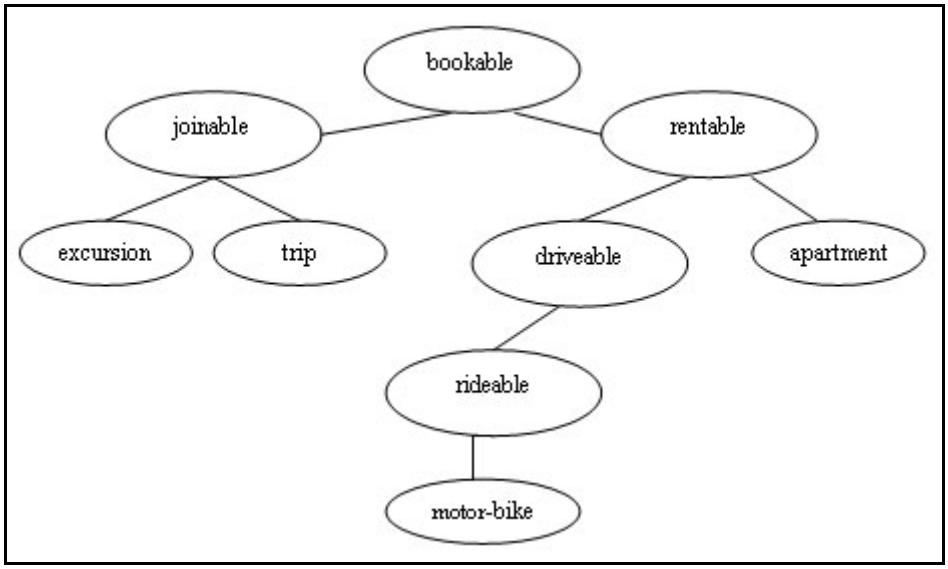


Abbildung 8: Konzepthierarchie nach Tabelle 1 aus CIMIANO ET AL. (2003:11).

Bei der Erstellung dieser Hierarchie wurden die Objekte der Verben hierarchisch angeordnet. Die Verben selbst bilden mit der Erweiterung „able“ abstrakte Konzepte, damit diese nach bestimmten lexikalischen Konzepten gruppiert werden können. Diese Grundüberlegung wird

zur Erstellung einer Taxonomie mit Hilfe der formalen Begriffanalyse benutzt [vgl. CIMIANO ET AL. 2003:11].

Um nun aus den Objekten und den dazugehörigen Attributen eine Taxonomie durch die formale Begriffsanalyse zu erhalten, werden Verb-Objekt Bindungen aus den Texten extrahiert, indem man die Grundformen der Objekte in FBA Objekte und die dazugehörigen Verben zusammen mit der Endung „able“ in Attribute wandelt.

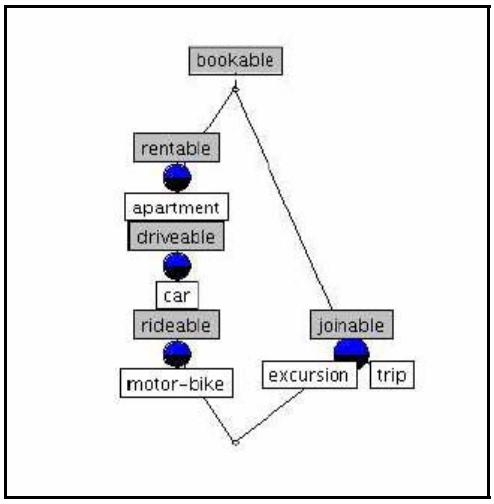


Abbildung 9: Konzepthierarchie nach formaler Begriffsanalyse in CIMIANO ET AL. (2003:11).

Wie man in Abbildung 9 sieht, sind Konzepte entstanden, die durch Verben kategorisiert sind, die diese charakterisieren (Beispiel alle Objekte des Verbs "rent" zu einer Klasse "rentable" zusammenfassen). Das Ergebnis von Versuchen zeigte, dass die selektive Bevorzugung von Verben größer ist, je seltener Nomen vorkommen, die direkte Objekte dieser Verben sind. Bei diesem Ansatz werden deshalb nur diese Verb-Term Paare als Paare von Attribut-Objekt angesehen [vgl. CIMIANO ET AL. 2003:13].

Der Vorteil dieses Ansatzes ist die Adaptivität – er kann auf jeden Korpus und jede Domäne angewandt werden [CIMIANO ET AL. 2003:10].]. Je spezifischer, technischer ein Korpus ist, desto stärker nähert man mit diesem Ansatz dem Niveau menschlicher Performance [CIMIANO ET AL. 2003:16].

4.5.3 Erstellung einer Ontologie mittels linguistischer Analyse

Wie weiter oben geschildert, kann ein typischer Ansatz beim Ontology Learning so aussehen, dass aus einem Korpus Terme extrahiert werden, um für eine Taxonomie potentielle Konzepte zu identifizieren. Der folgende Ansatz von BUITELAAR ET AL. (2004) verläuft ähnlich, jedoch wird hier mehr auf die Entwicklung von Ontologien mit Hilfe einer statistischen Analyse gezielt. Dieser Ansatz wurde mit OntoLT implementiert. OntoLT ist ein Plugin für den Ontologieeditor Protégé. Auf beide wird bei der Erklärung der Vorgehensweise des Versuchs in Kapitel 5.3.2 genauer eingegangen.

Konzepte und Attribute werden bei diesem Ansatz automatisch aus einer annotierten, domänen-spezifischen Textkollektion extrahiert und in eine Ontologie integriert, indem Mapping-Regeln angewendet werden. Dafür werden Mapping-Regeln definiert, die festlegen, wie linguistische Bestandteile eines Korpus auf Konzepte oder Attribute abgebildet werden können. Ein wichtiger Aspekt dieses Ansatzes ist die semi-automatische Vorgehensweise. Dies wird dadurch deutlich, dass durch Interaktion mit einem Experten die Mapping-Regeln erweitert und Konzepte daraufhin manuell ausgewählt werden.

Eine Regel besteht aus Bedingungen und Operatoren. Die Bedingungen bzw. Voraussetzungen beschreiben linguistische Spezifikationen, die von einem Kandidaten eingehalten oder erfüllt werden müssen, wie zum Beispiel, dass die Kandidaten Prädikate, Subjekte oder direkte Objekte sein sollen. Mit den Operatoren wird ausgedrückt, auf welche Art und Weise die Ontologie erweitert werden soll, also ob Kandidaten als Oberklasse oder als Attribut in eine Ontologie eingefügt werden sollen [vgl. BUITELAAR ET AL. 2003]. Die Mapping-Regeln werden über eine formale Bedingungssprache ausgedrückt, wodurch linguistische Einheiten in den annotierten Dokumenten gezielt ausgewählt werden können. Dabei werden Bedingungen ausgedrückt, die bei der Extraktion von Konzeptkandidaten beachtet werden sollen, wie z.B. dass nur Nomen (head noun) bei der Auswahl berücksichtigt werden sollen. Die Bedingungen werden dann auf das annotierte Korpus angewandt. Je nachdem welche linguistischen Vorbedingungen aufgrund der festgelegten Regel von den einzelnen Satzbestandteilen erfüllt werden, kommen im Folgenden passende Operatoren zum Einsatz, die potentielle Kandidaten für ein Konzept oder Attribut zu nominieren, die dann vom Benutzer ausgewählt werden können. Die ausgewählten Kandidaten können anschließend automatisch zu einer neuen oder bereits bestehenden Ontologie hinzugefügt werden [vgl. BUITELAAR ET AL. 2004:32]. Ein Beispiel für eine Mapping-Regel könnte

folgendermaßen aussehen: die Regel wählt z.B. alle Terme aus, die zur linguistischen Klasse Nomen gehören. Daraufhin wird dem Benutzer eine automatisch generierte Liste mit Konzepten präsentiert, die die extrahierten Nomen enthält [vgl. BUITELAAR ET AL. 2003].

Zur Erweiterung der Mapping-Regeln um relevante Konzepte kann eine statistische Analyse durchgeführt werden. Diese basiert auf dem Einsatz der so genannten „chi-square"-Funktionen in AGIRRE ET AL. (2001). Diese Funktion berechnet eine Relevanzpunktzahl, indem die Frequenzen in einem Domänenkorpus unter Berücksichtigung der Frequenzen im Referenzkorpus verglichen werden. Beispielweise könnte das Britisch National Corpus, welches Informationen über die Frequenzen englischer Wörter im heutigen Sprachgebrauch enthält, verwendet werden. Auf diesem Wege wird der Gebrauch von Wörtern in bestimmten Domänen dem Gebrauch von Wörtern im Allgemeinen gegenüber gestellt [vgl. BUITELAAR ET AL. 2004:38]. Folglich wird dem Benutzer eine Liste mit automatisch generierten Termen vorgeschlagen, so dass ein Benutzer Konzepte für eine Ontologie auf Basis der so extrahierten Terme auswählen kann. Die Auswahlmöglichkeit und die Vorschläge von Konzepten machen die Interaktivität dieses Ansatzes deutlich.

Dadurch, dass Mapping-Regeln eingesetzt und erweitert werden, bleibt domänen-spezifisches Wissen über die Zusammenhänge von Wörtern, über morphologische und syntaktische Strukturen mit der Ontologie verbunden, wovon man bei der Erweiterung oder bei der Pflege der Ontologie profitiert [vgl. BUITELAAR ET AL. 2004:32]. Für diesen Zweck kann man Mapping-Regeln für alle möglichen XML-Elemente der linguistischen Annotation generieren, beschränkt auf die Worte, die nach der chi-square-Analyse ausgewählt wurden. Jedoch benötigt man immer noch die Interaktionen eines Experten, um die Operatoren, die mit den generierten Bedingungen für die festzulegenden Regeln verbunden sind, einsetzen zu können.

In diesem Kapitel wurden diverse Ansätze und Vorgehensweisen von maschinellen Lerntechniken im Bezug auf Ontologien vorgestellt. Nachdem die wichtigsten Ansätze gezeigt wurden, soll im nächsten Kapitel ein Teil dieser Lerntechniken in Bezug auf das virtuelle Bibliotheksregal MyShelf angewendet werden.

5 Maschinelles Lernen von Ontologien für MyShelf

In den vorhergehenden Kapiteln wurden Grundlagen und Ansätze des maschinellen Lernens von Ontologien erläutert. Im Folgenden Abschnitt dieser Arbeit soll die praktische Anwendung einiger dieser Ansätze beschrieben werden.

5.1 Zielsetzung

Wie in Kapitel 1 bereits beschrieben, haben sich mehrere Magisterarbeiten mit der Problematik der semantischen Heterogenität an der UB Hildesheim beschäftigt. Die Möglichkeit die Perspektiven der Klassifikation auf den Bestand zu wechseln (vgl. HEINZ 2003) soll um weitere Perspektiven zum Ontology Switching erweitert werden, unter anderem, um die Berücksichtigung elektronischer Medien zu ermöglichen [vgl. KÖLLE ET AL. 2004]. Die Möglichkeit die von HEINZ (2003) bereits errichtete Auswahl verschiedener Perspektiven („Brillen“) auf den betroffenen Bestand wechseln zu können, soll durch die Erstellung zusätzlicher Ontologien unter Berücksichtigung digitaler Bestände getestet werden.

Für das maschinelle Lernen von Ontologien werden zwei Tools verwendet. Zum einen das von MAEDCHE & STAAB (2004) beschriebene KAON TextToOnto Version 1.0[2], das vom Institut für Angewandte Informatik und Formale Beschreibungsverfahren (AIFB) in Karlsruhe entwickelt wurde, zum anderen OntoLT Version 1.0[3], welches am DFKI (Deutsches Forschungszentrum für Künstliche Intelligenz) in Saarbrücken von BUITELAAR ET AL. (2004) entwickelt wurde. OntoLT ist ein Plugin für den bekannten Ontologieeditor der Universität Standford Protégé[4].

Ausgangspunkt für die Erstellung einer Ontologie stellen Textkorpora dar. Hierfür wurden zum einen Internetsites, die im Rahmen der Magisterarbeit von WILHELM (2004) für MyShelf erschlossen wurden und zum Großteil aus dem Bereich Computerlinguistik stammen,

[2] http://sourceforge.net/projects/texttoonto

[3] http://olp.dfki.de/OntoLT/OntoLT.htm

[4] http://protege.stanford.edu/

berücksichtigt. Zum anderen wurden PDF-Dokumente von CiteSeer [NEC RESEARCH INSTITUTE 2004] für die Bereiche Information Retrieval und Computerlinguistik erschlossen.

Im Folgenden sollen die Erschließung der Korpora sowie die Anwendung verschiedener Lernansätze auf diese mit den oben genannten Tools beschrieben werden. Ziel ist es, aus den erschlossenen Korpora Ontologien in Form von Konzepthierarchien maschinell lernen zu lassen.

5.2 Erschließung der Korpora

Um eine Ontologie maschinell lernen zu lassen, wird ein Korpus benötigt, welches die Ausgangsbasis für die Lernvorgänge bildet. Zum einen müssen der Lernalgorithmus und die Vorgehensweise entsprechend erfolgreich arbeiten, damit aus einem Korpus hinreichende Konzepte extrahiert werden können. Zum anderen hat allerdings natürlich das Korpus selbst einen großen Einfluss auf das Lernergebnis, da er die Basis zum Lernen darstellt.

5.2.1 Auswahl der Dokumente

Für die automatische Erstellung einer Ontologie für MyShelf werden im Folgenden zwei Korpora verwendet. Zum einen werden Dokumente auf Basis der von WILHELM (2004) ermittelten Links erschlossen, zum anderen werden neue Dokumente von CiteSeer [NEC Research Institute 2004] hinzugefügt. Da das Tool TextToOnto derzeit nur englische Korpora verarbeiten kann, werden für die Erschließung der Korpora ausschließlich englischsprachige Dokumente in Erwägung gezogen. Bei der Auswahl der beiden Korpora lag im Vordergrund, dass es elektronisch zugängliche Textquellen sein mussten, die besser zur Aufbereitung und Verarbeitung am Computer geeignet waren als ein Bücherbestand. Auch wurden die Korpora auf nur zwei Teilbereiche der Informationswissenschaft beschränkt, um den Aufwand in einem angemessenen Rahmen zu halten.

WILHELM (2004) sammelte Internetlinks, die nach diversen Kriterien ausgewählt und dementsprechend in die Hanke-Klassifikation übertragen wurden. Somit finden sich in der Datenbank ein Großteil der Links zum Thema Sprachtechnologie mit den Bereichen Computerlinguistik, Maschinelle Übersetzung, Mensch-Maschine-Interaktion, Multilinguale Informationssysteme und Translation Memory [vgl. WILHELM 2004:16f]. Dabei wurden deutsch- und englischsprachige Internetlinks aus den Bereichen Sprachtechnologie und Information Retrieval in die Datenbank aufgenommen. Für das Korpus, das auf den gesammelten Links von WILHELM (2004) beruht, sollen, wie bereits erwähnt, nur englischsprachige Sites aus den Teilgebieten Sprachtechnologie und Information Retrieval berücksichtigt werden. Diese Dateien bilden somit das erste Korpus.

Das zweite Korpus wurde aus Dokumenten von CiteSeer erschlossen werden. Da die Dokumente des Gesamtkorpus letztendlich die Teilbereiche Sprachtechnologie und Information Retrieval abdecken sollen, bietet sich CiteSeer hierfür sehr gut an. CiteSeer ist eine digitale Bibliothek und Suchmaschine, die wissenschaftliche Publikationen online zugänglich macht, indem Forschungsartikel als PostScript- und PDF-Dateien bereitgestellt werden. CiteSeer wurde vom NEC Research Institute entwickelt und wird in Kooperation mit der Pennsylvania State University [NEC RESEARCH INSTITUTE 2004a] angeboten. CiteSeer bietet einen Online-Suchindex für wissenschaftliche Veröffentlichungen aus den Bereichen Computer Science und Information Science.

Abbildung 10: CiteSeer-Suchmaske [vgl. NEC RESEARCH INSTITUTE 2004].

Das Ziel von CiteSeer ist es, wissenschaftliche Artikel online leichter zugänglich zu machen, da diese Artikel meistens im Internet schlecht erreichbar und unorganisiert sind. CiteSeer erschließt automatisch Artikel aus dem Internet. Es werden PostScript Dateien und PDF-Dokumente erschlossen, welche darauf geprüft werden, ob es sich um wissenschaftliche Artikel handelt [vgl. LAWRENCE ET AL. 1999:68]. CiteSeer eröffnet die Möglichkeit, dass man die digitale Bibliothek über eine Keyword-Suche durchsuchen kann (siehe Abbildung 10).

5.2.2 Erschließung der Korpora

Wie eben beschrieben sollen die Korpora nur englischsprachige Dokumente beinhalten, die dann als reine Textdokumente verwendet werden sollen. Um eine ausreichend große Ausgangsbasis zu schaffen, wurde ein Minimum von insgesamt 1000 Dokumenten pro Korpus festgesetzt.

Datenbank des Virtuellen Wegweisers Informationswissenschaft

Auf Basis der Datenbank von WILHELM (2004) wurden die englischsprachigen Internetlinks aus den Bereichen Sprachtechnologie und Information Retrieval ermittelt. Die Verteilung der Internetlinks in den jeweiligen Kategorien und den dazugehörigen Unterkategorien bezüglich deutscher und englischer Sites verhält sich wie in Tabelle 2 dargestellt.

Eine genauere Darstellung inklusiver grafischer Darstellung dieser Dokumentenverteilung befindet sich im Anhang II.

Wie man sieht, liegt der Schwerpunkt auf Sprachtechnologie. Der Bereich Information Retrieval muss also verstärkt mit CiteSeer-Dokumenten abgedeckt werden. Die so ermittelten URLs wurden daraufhin mit Anawave WebSnake Version 1.23[5] heruntergeladen. Dieses Programm ist ein Downloadmanager, welches es ermöglicht ganze Websites inklusiver ihrer Strukturen herunterzuladen.

[5] http://www.anawave.com

Kategorie(ID)	Kategoriename	Anzahl Links	Deutsch	Deutsch/ Englisch	Englisch
Alle Kategorien:		**388**	**121**	**17**	**250**
36	Grundlagen/ Theorie	1	1	0	0
125	**Information Retrieval**	**9**	**1**	**0**	**8**
160	Human Computer Interaction	114	43	5	66
167	**Sprachtechnologie**	**260**	**76**	**12**	**172**
188	Wissensvermittlung/ Informations- und Dokumentationsstellen	2	0	0	2
218	Künstliche Intelligenz	2	0	0	2

Tabelle 2: Sprachverteilung der Internetsites von WILHELM (2004).

Da die betroffenen Websites sich allerdings auf mehrere Ebenen in die Tiefe erstrecken, wurde festgelegt, dass maximal drei Ebenen berücksichtigt werden und keine Links zu externen und damit eventuell themenfremden Seiten weiterverfolgt werden sollten. Mit Hilfe von WebSnake wurden dann ausschließlich HTML- und Text-Dokumente bis zur dritten Ebene der Gesamtsite erschlossen. Auch wurde zusätzlich ein Maximum von 250 Dateien pro Site festgelegt. Die erschlossenen HTML-Dokumente sollen daraufhin in reine Textdokumente konvertiert werden. Die heruntergeladenen Seiten werden mit dem Tool HTMLAsText Version 1.01[6] in reine Textdateien konvertiert. Anschließend werden diese neuen Textdokumente manuell überprüft und fehlerhafte aussortiert.

Dokumente von CiteSeer

Um die bereits beschriebenen Dokumente von CiteSeer herunterzuladen, wurde ein Java-Tool entwickelt. MyCrawler Version 1.0 (siehe Abbildung 11) ermöglicht es, Dokumente von CiteSeer automatisch zu erschließen.

[6] http://www.nirsoft.net

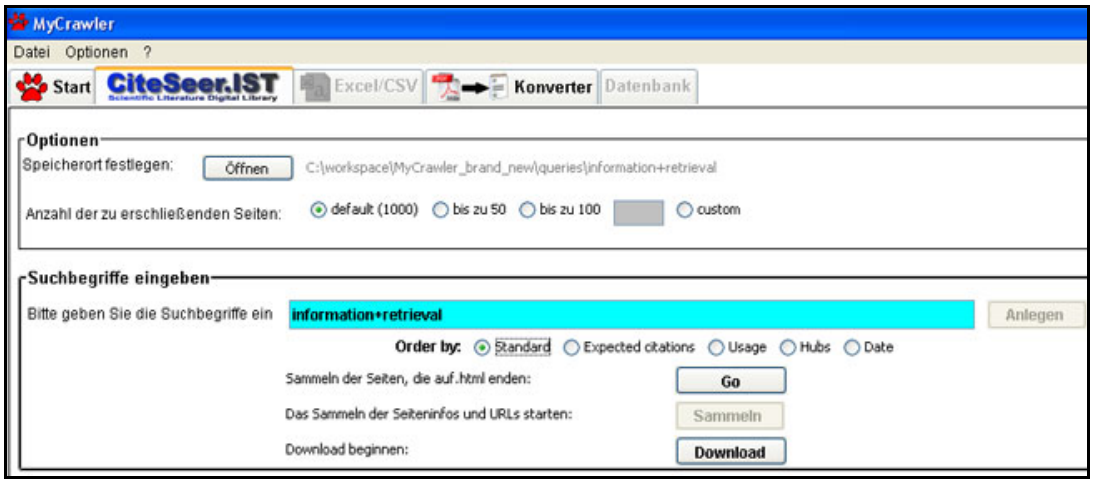


Abbildung 11: Eingabemaske für CiteSeer in MyCrawler.

Um einen CiteSeer Artikel herunterzuladen, kann man über eine Suchmaske nach diesem Artikel suchen. Verfolgt man nun einen der Treffer, so gibt es auf der folgenden Eingabemaske die Möglichkeit, die Artikel als PostScript- oder PDF-Datei herunterzuladen. Möchte man eine große Anzahl an Dokumenten weitgehend automatisch erschließen, kann man mit MyCrawler das Ergebnis einer Suche automatisch als PostScript-Dateien herunterladen. Zusätzlich zu der Möglichkeit Dokumente zu suchen, bietet CiteSeer auch ein Verzeichnis an, welches die indizierten Publikationen zu verschiedenen Kategorien beinhaltet. Dieses Verzeichnis umfasst viele Bereiche von Computer Science, wie z.B. Agents, Artificial Intelligence, Information Retrieval, Machine Learning etc. sowie dazugehörige Unterverzeichnisse. Es wird automatisch erstellt, so dass einige Artikel sowohl in falsch zugeteilten Kategorien oder mehrmals in verschiedenen zu finden sind. Das Verzeichnis soll demnach nur eine Einstiegshilfe beim Browsing sein und ist deswegen nicht unbedingt verbindlich [vgl. NEC RESEARCH INSTITUTE 2004b].

Mit MyCrawler sollen Unterverzeichnisse des genannten Verzeichnisses und Dokumente zu diversen Suchbegriffen erschlossen werden. Es werden also die PostScript-Dokumente heruntergeladen. Die somit erschlossenen PostScript-Dokumente werden dann mit Jaws PDFCreator v3.3 in PDF-Dateien konvertiert und anschließend mit Hilfe von MyCrawler oder PDF2Text v3.0[7] in reine Textdokumente konvertiert. TextToOnto bietet zwar auch die Verarbeitung von PDF-Dateien an, allerdings ist derzeit nur die Verarbeitung von HTML-

[7] http://www.verypdf.com/pdf2txt/pdf2txt.htm

und Textdokumenten implementiert. Anschließend werden fehlerhafte Dokumente manuell aussortiert.

Die genaue Vorgehensweise und Funktionalität von MyCrawler kann dem Handbuch in Anhang IX entnommen werden.

5.2.3 Ergebnis der Erschließung

Nach den bereits beschriebenen Vorgehensweisen wurden für beide Korpora Dokumente gesammelt, die nun als reine Textdateien zur Verfügung stehen. Eine Übersicht soll das Ergebnis der Sammlung der Dokumente für die beiden Teilkorpora zeigen.

Verteilung der Dokumente der Datenbank-Links

Insgesamt wurden 8300 HTML-Dokumente gesammelt, wovon nach der Konvertierung und Aussortierung 6182 Textdokumente zur Verfügung stehen. Beim Download wurden HTML- und PDF-Dateien erschlossen, wobei die überwiegende Mehrheit HTML-Dateien ausmachen. Die konvertierten Dokumente wurden anschließend nach der Link-ID aus der Datenbank benannt. So wird zum Beispiel die Datei index.html in DB-002_index.html umbenannt. Die genaue Auflistung der erschlossenen Sites kann in der Excel-Datei DB_EnglischeSites.xls auf der beiliegenden DVD 1 –Anlage 3.7 eingesehen werden.

Verteilung der Dokumente von CiteSeer

Mit MyCrawler wurden insgesamt 2270 Dokumente zu diversen Suchbegriffen und aus bereits angelegten Verzeichnissen erschlossen. Dabei wurden die in Tabelle 3 dargestellten Suchbegriffe und Verzeichnisse verwendet.

Es bleibt zu erwähnen, dass Dateien doppelt oder sogar mehrfach erschlossen wurden, was darauf zurückzuführen ist, dass beim Download aus den CiteSeer eigenen Verzeichnis mehrere Dokumente unter verschiedenen Kategorien zu finden sind und dass Artikel bei verschiedenen Varianten von Suchbegriffen, die sich ähnlich oder aus dem gleichen wissenschaftlichen Teilgebiet sind, ebenso mehrmals vorkommen.

ID	Kategorie	Bereich	Ordnername	Anzahl Textdokumente
CS-CL01	CiteSeer	Computerlinguistik	computer+aided+translation	11
CS-CL02	CiteSeer	Computerlinguistik	computer+linguistics	66
CS-CL03	CiteSeer	Computerlinguistik	machine+translation	178
CS-CL04	CiteSeer	Computerlinguistik	NaturalLanguageProcessing	168
			Gesamt:	**423**
CS-IR01	CiteSeer	Information Retrieval	Classification	129
CS-IR02	CiteSeer	Information Retrieval	data+mining	185
CS-IR03	CiteSeer	Information Retrieval	DigitalLibraries	129
CS-IR04	CiteSeer	Information Retrieval	Extraction	115
CS-IR05	CiteSeer	Information Retrieval	Filtering	135
CS-IR06	CiteSeer	Information Retrieval	information+extraction	145
CS-IR07	CiteSeer	Information Retrieval	information+filtering	58
CS-IR08	CiteSeer	Information Retrieval	information+retrieval	290
CS-IR09	CiteSeer	Information Retrieval	InformationRetrieval	158
CS-IR10	CiteSeer	Information Retrieval	Metasearch	117
CS-IR11	CiteSeer	Information Retrieval	Retrieval	127
CS-IR12	CiteSeer	Information Retrieval	SearchEngines	127
CS-IR13	CiteSeer	Information Retrieval	WorldWideWeb	132
			Gesamt:	**1847**
			Insgesamt:	**2270**

Tabelle 3: Verwendete Suchbegriffe und Kategorien von CiteSeer mit MyCrawler.

Um dennoch identifizieren zu können, welches Dokument zu welchem Downloadvorgang und welcher Suchkategorie gehört, wurden alle konvertierten Dokumente mit der ID aus Tabelle 3 versehen.

Dieses Korpus soll im Folgenden die Basis für den Lernvorgang bilden. Das Gesamtkorpus inklusive Originaldateien und konvertierten Textdokumenten ist auf DVD 1 und DVD 2 jeweils unter Anlage 2 zu finden. Die Auflistung der Suchbegriffe ist dem Anhang III zu entnehmen. Die Verteilung der Dokumente über die Suchkategorien ist grafisch auch in Anhang IV dargestellt.

5.3 Vorgehensweise

Zur Verarbeitung der erschlossenen Korpora werden in Folgenden zwei Tools verwendet, die auf oben bereits erwähnten Ansätzen basieren. Dabei sollen deren Aufbau und

Funktionsweise kurz erklärt und die für den Versuch relevanten Einstellungen dargestellt werden. Es sollen in den Versuchen Konzepthierarchien gelernt werden und nicht automatische Klassifizierung von Instanzen oder Erstellung von Attributen durchgeführt werden.

5.3.1 KAON TextToOnto

KAON TextToOnto des AIFB Karlsruhe ist ein in Java implementiertes Lernsystem für Ontologien. TextToOnto ist in das Ontologiemanagementsystem KAON eingebettet. Wie oben bereits erwähnt, schildern MAEDCHE & STAAB (2004) eine Architektur eines Lernsystems für Ontologien, welche auf TextToOnto übertragen wurde. Dabei stellt KAON diverse Möglichkeiten zur Bearbeitung einer Ontologie zur Verfügung. TextToOnto (siehe Abbildung 12) beinhaltet Möglichkeiten zur Vorverarbeitung von Texten, sowie Algorithmen zur Extraktion einer Ontologie. Die Architektur von TextToOnto erlaubt es, Methoden und Techniken zur maschinellen Erstellung einer Ontologie zu verwenden und ebenfalls Ontologien manuell zu bearbeiten. KAON TextToOnto ist Open-Source und kann auf der Internetseite von KAON heruntergeladen werden [vgl. MAEDCHE; STAAB 2004:183ff].

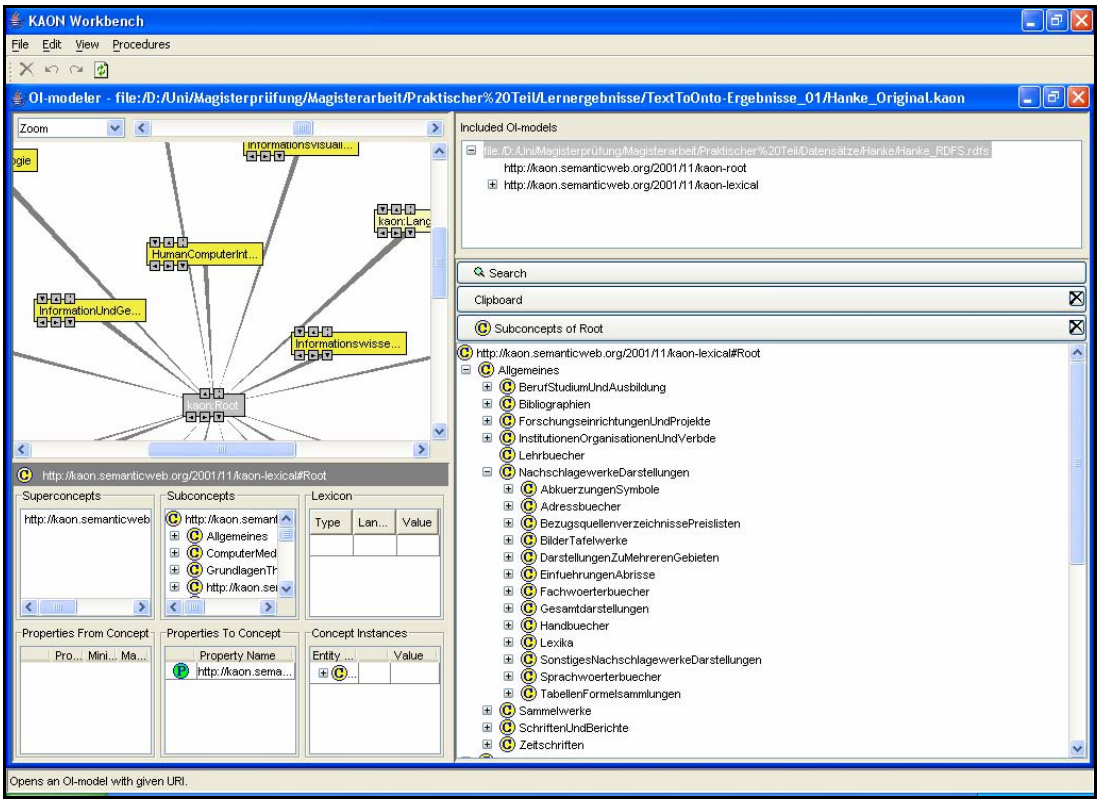


Abbildung 12: KAON TextToOnto.

KAON verwendet Modelle von Ontologien, die als OI-Modelle (Ontology-Instance Model) bezeichnet werden und über eine grafische Benutzeroberfläche bearbeitet werden können (siehe Abbildung 13). Die Modelle enthalten alle Informationen zu den Konzepten, deren Attributen und Instanzen [vgl. MAEDCHE; STAAB 2004:176].

Um Ontologien mit TextToOnto maschinell lernen zu lassen, kann man in TextToOnto ein domänen-spezifisches Korpus festlegen, auf welches dann verschiedene Algorithmen zur Extraktion einer Ontologie angewendet werden können. Mit Hilfe von integrierten Werkzeugen zur linguistischen Analyse kann diese Kollektion analysiert und linguistisch aufbereitet werden, so dass Terme aus dem annotierten Text extrahiert werden können. Die einzelnen Bestandteile einer Ontologie werden hierbei in einem OI-Modell angelegt, welches dann die Konzepthierarchie enthält.

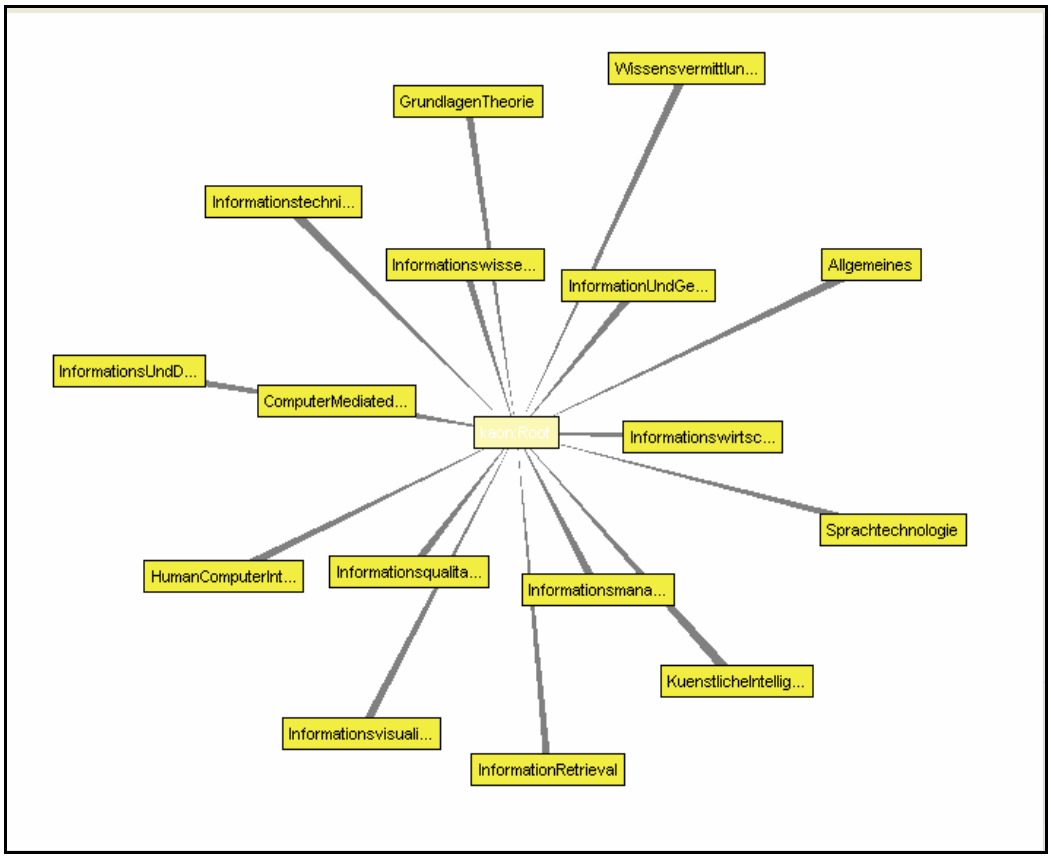


Abbildung 13: Grafische Darstellung von Konzepten in TextToOnto.

Für den Versuch wurde der in TextToOnto enthaltene TaxoBuilder (siehe Abbildung 14) verwendet. Der TaxoBuilder bietet die Möglichkeit, über verschiedene Ansätze Konzepthierarchien aus einem Korpus automatisch aus den Termen eines Korpus zu erstellen, welche dann in ein leeres OI-Modell integriert werden. Hierbei können zwei Methoden verwendet werden. Zum einen können Konzepte über eine formale Begriffsanalyse (siehe Kapitel 4.5.2) extrahiert werden. Hierbei wird der Ansatz von CIMIANO ET AL. (2003) angewendet. Wie oben bereits erklärt (siehe 4.5.2) geht man davon aus, dass ein Verb starke Selektionsbeschränkungen voraussetzt, so dass eine Konzepthierarchie dadurch abgeleitet werden kann, dass die Beziehungen von Verben zu ihren Objekten aufgrund dieser Selektionsbeschränkungen bestimmt werden. Dabei kann man auswählen, ob alle Verben des Korpus als potentielle Merkmale berücksichtigt werden sollen, oder ob die Verben insgesamt auf weniger Konzepte abgebildet werden sollen, nämlich auf die lexikografischen Klassen in WordNet, welches semantische Beziehung von Konzepten enthält.

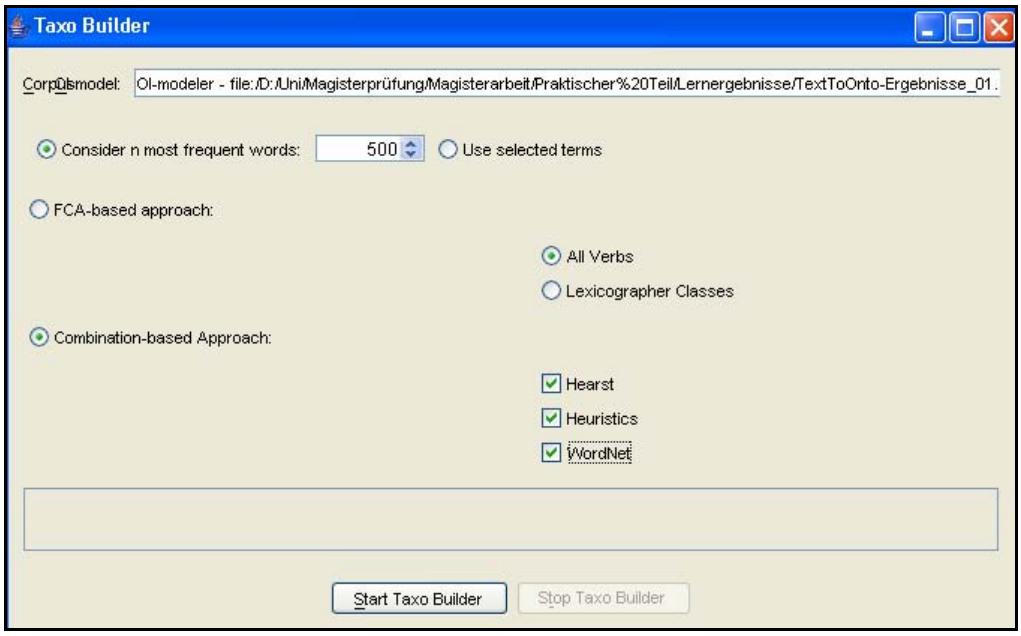


Abbildung 14: TaxoBuilder in TextToOnto.

Die zweite Möglichkeit verwendet die Kombination heterogener Beweisquellen (siehe Kapitel 4.5.1). Dabei werden Hearst-Pattern, WordNet und verschiedene Heuristiken verwendet und miteinander kombiniert. Dabei werden alle im Korpus vorkommenden Nominalphrasen mit nur einer Vergleichsphrase in eine is-a-Beziehung gestellt. Zur Extraktion von is-a-Beziehungen in einem Korpus sollen bei diesem Ansatz zwei Terme auf eine Hyponymie-Beziehung unter Berücksichtigung von WordNet untersucht werden. Da Wörter mehrere Bedeutungen haben können, werden hier alle Bedeutungen eines Terms zur Paarbildung herangezogen. Bei diesem Ansatz arbeitet TextToOnto mit der WordNet Version 1.7.1[8]. Auch wird die bereits erläuterte Heuristik „vertikaler Relationen“ verwendet. Hierbei werden zwei Terme auf eine is-a-Beziehung nach der beschriebenen Vorgehensweisen untersucht. Mit Hilfe der Heuristiken entsteht diese Beziehung dadurch, dass zu dem ersten Term ein zweiter in Bezug gestellt wird, der erste jedoch mit Adjektiven oder anderen Begriffen modifiziert wird, so dass Mehrwortterme herangezogen werden können.

Die linguistische Aufbereitung der Korpusdaten, die dann durch die enthaltenen Algorithmen verarbeitet werden sollen, kann entweder über GATE oder über einen Standardprozessor zur Verarbeitung englischer Texte erfolgen. Das auf Java basierende GATE (General Architecture for Text Engineering) wurde von der Universität Sheffield entwickelt und stellt ein NLP-

[8] ftp://ftp.cogsci.princeton.edu/pub/wordnet/1.7.1

System dar, mit welchen linguistische Informationen aus Texten extrahiert werden und auch Texte annotiert werden können [NLP GROUP 2005]. In den hier durchgeführten Versuchen wurde der in TextToOnto integrierte Standardprozessor zur Aufbereitung des Korpus verwendet. Dieser extrahiert relevante Terme und verwendet reguläre Ausdrücke auf Part-Of-Speech Tags, welche einem Text jedes Satzteils linguistisch kennzeichnen [vgl. TEXTTOONTO 2003].

Ziel ist es, aus einem domänen-spezifischen Korpus eine Ontologie erstellen zu lassen. Dabei wird TaxoBuilder folgendermaßen eingesetzt: als Eingabedaten zum Lernen der Ontologie werden zwei Korpora zur Verfügung gestellt. Zum einen erschlossene Internetsites, die in der Datenbank von WILHELM 2003 enthalten sind, und zum anderen PDF-Dokumente von CiteSeer für die Bereiche Computerlinguistik und Information Retrieval. Die beiden Korpora wurden mit den beiden angebotenen Ansätzen verwendet, wobei teilweise mit kleineren Teilen der Korpora und weniger Textdokumenten gearbeitet werden musste, da die Größe der Korpora teilweise für die Verarbeitung mit TextToOnto problematisch war (siehe Kapitel 5.4).

TextToOnto - TaxoBuilder			
	Korpora	**Anzahl Konzeptkandidaten**	**Parameter**
Kombinations-basierter Ansatz	▪ Datenbank ▪ CiteSeer	▪ Frei wählbar	▪ Hearst-Pattern ▪ Heuristik „vertikaler Relationen“ ▪ WordNet
Formale Begriffsanalyse	▪ Datenbank ▪ CiteSeer	▪ Frei wählbar	▪ Alle Verben im Korpus ▪ Lexikografische Klassen

Tabelle 4: Übersicht über die Parameter beim TaxoBuilder.

Um den Einfluss einzelner Parameter auf das Lernergebnis feststellen zu können, wurden bei der Versuchsanordnung verschiedene Parametereinstellungen ausprobiert und kombiniert. Dabei konnte für den Ansatz der formalen Begriffsanalyse und den kombinations-basierten Ansatz verschiedene Parameter gewählt werden. Die verschiedenen Parameter und Korpora werden in einer Übersicht in Tabelle 4 dargestellt. Für beide Ansätze konnte festgelegt werden, wie viele der am häufigsten vorkommenden Terme im Korpus für die Verarbeitung mit den beiden Ansätzen verwendet werden sollen.

Den genauen Aufbau aller Versuchsanordnungen können dem Anhang VI entnommen werden.

5.3.2 OntoLT

Ein weiterer Versuch wurde mit OntoLT (siehe Abbildung 15) durchgeführt. OntoLT ist ein Java-basiertes Plugin für den Ontologieeditor Protégé[9]. OntoLT benutzt die bereits in 4.5.3 erläuterte Extraktion von Termen aus einem Korpus mit Mapping-Regeln. Diese legen fest, auf welche Art linguistische Einträge aus einem annotierten Korpus auf Konzepte oder Attribute abgebildet werden können. Eine Regel besteht dabei aus Bedingungen und Operatoren. Die Bedingungen bzw. Voraussetzungen beschreiben linguistische Spezifikationen, die von einem Kandidaten eingehalten oder erfüllt werden müssen. Dabei stehen im Plugin einige vordefinierte Regeln zur Verfügung, jedoch kann der Benutzer auch neue Regeln definieren. Die Regeln definieren Bedingungen, die bei der Extraktion von Konzeptkandidaten beachtet werden sollen, wie z.B. dass nur Nomen bei der Auswahl berücksichtigt werden sollen. Diese werden in XPATH, einer Sprache zur Adressierung von Teilen eines XML-Dokuments, ausgedrückt.

Diese Bedingungen werden auf das Korpus angewandt. Im Korpus werden Satzteile mit XML-Tags versehen, die die linguistische Rolle (semantische Muster) der einzelnen Satzbestandteile beschreiben [vgl. BUITELAAR ET AL. 2004:43]. Derartig annotierte Dokumente können dann in OntoLT importiert und weiterverarbeitet werden. Alle Dokumente eines Korpus müssen zur Verarbeitung mit OntoLT spezifisch annotiert werden. Da die linguistische Annotation in OntoLT selbst nicht integriert ist, erfolgt die Annotation der Korpora über SCHUG (Shallow and Chunk based Unification Grammar) [vgl. DECLERCK 2000], einem regelbasierten System zur Analyse deutscher und englischer Texte. SCHUG beinhaltet linguistische Werkzeuge, die Sätze in einer XML-Struktur linguistisch und semantisch beschreiben.

[9]OntoLT wird mit Protégé 2000 in der Version 1.8 verwendet.

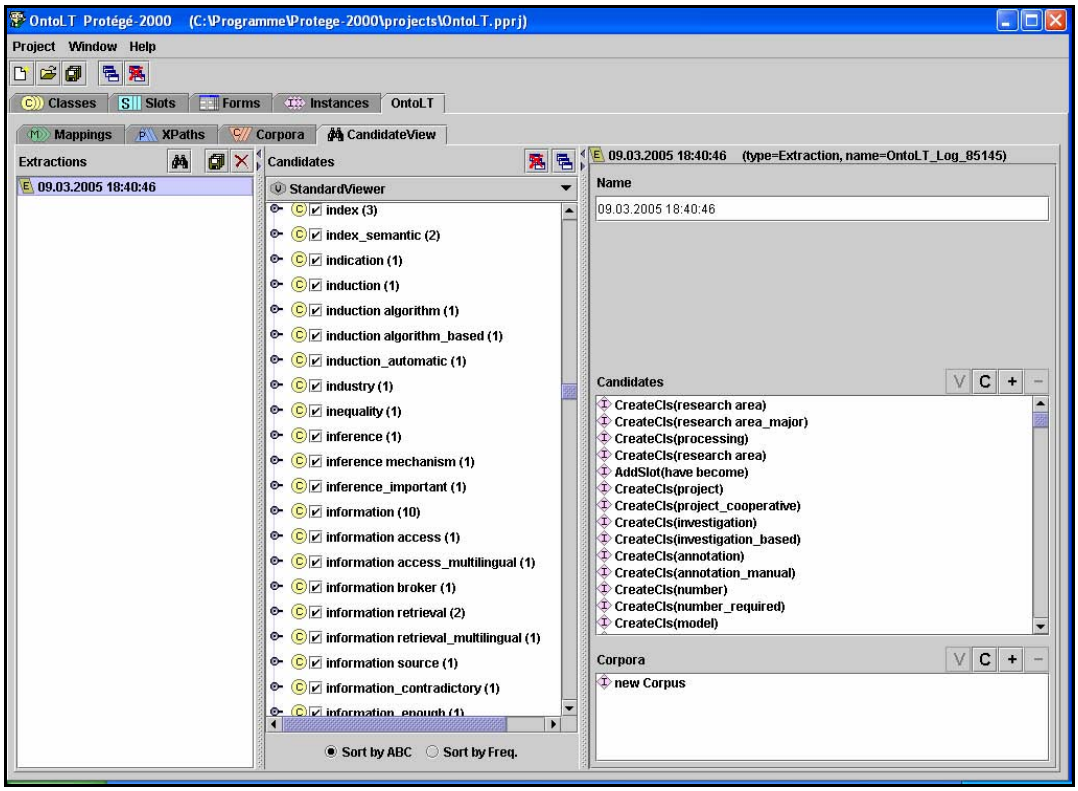


Abbildung 15: OntoLT – Konzeptextraktion.

Die Annotation integriert dabei mehrere Ebenen linguistischer und semantischer Analyse. SCHUG stellt die Satzbestandteile nach einer linguistischen Analyse in einer XML-Struktur dar. Mit dieser einheitlichen Darstellung der Dokumente sind spezifische linguistische Informationen strukturiert zugänglich und können dann über die Mapping-Regeln gezielt angesprochen werden. Die Kennzeichnung linguistischer Einheiten durch SCHUG richtet sich an einen Abschnitt des Dokuments, der Informationen zum Part-Of-Speech Tagging beinhaltet. Auch morphologische Informationen über Flexionen und Kompositazerlegung, sowie die Untersuchung von Phrasenstruktur werden in die Annotation integriert [vgl BUITELAAR ET AL. 2004:34]. Ein Beispielabschnitt eines mit SCHUG annotierten Satzes steht im Anhang V zur Verfügung. Dieser Ausschnitt zeigt die unterschiedlichen Ebenen, die Informationen über Part-Of-Speech Tagging, die Phrasenstrukturgrammatik und die syntaktische Struktur enthalten. Die Vorgehensweise von OntoLT ist in Abbildung 16 nochmals im Überblick dargestellt.

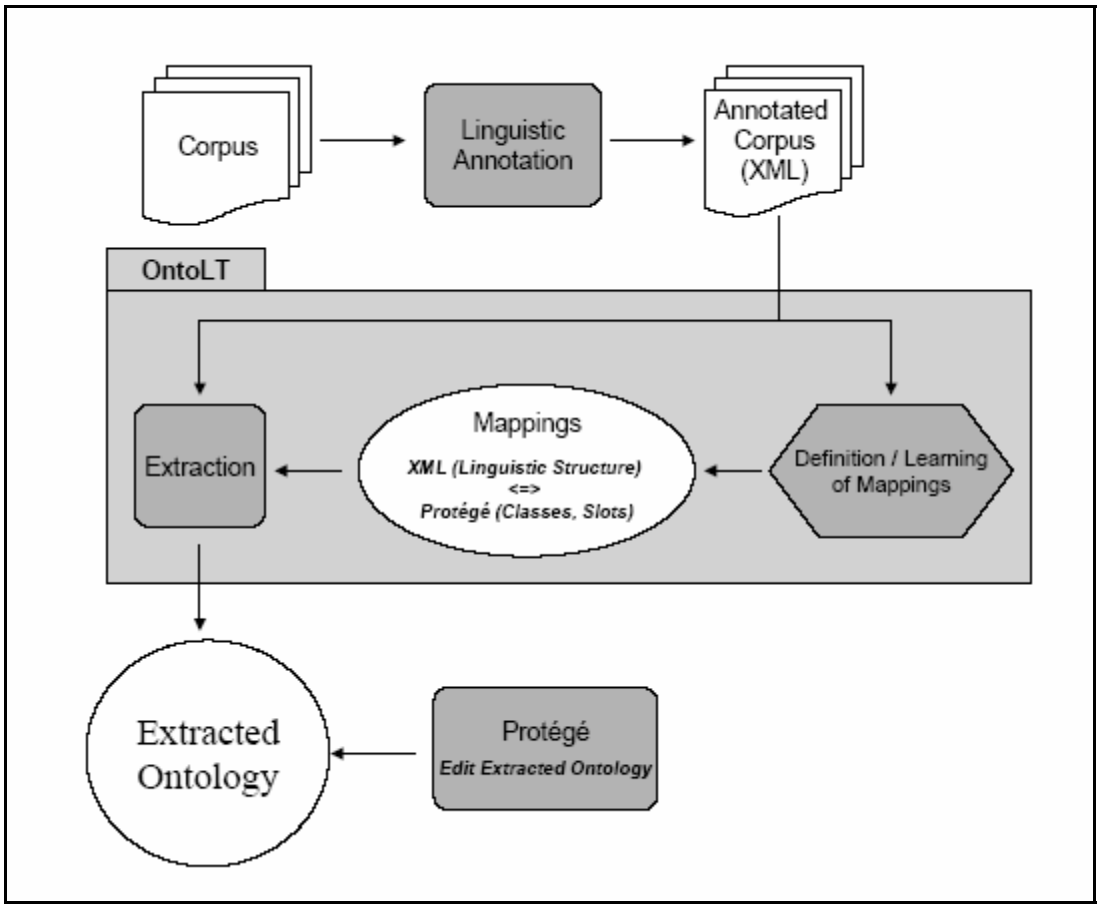


Abbildung 16: Überblick über den Ansatz von OntoLT in BUITELAAR ET AL. (2004:33).

Obwohl SCHUG derzeit nicht frei verfügbar ist, konnte mit freundlicher Unterstützung von Paul Buitelaar und Alexander Schutz vom DFKI Saarbrücken ein Teil der Dokumente des CiteSeer-Korpus aus dem Bereich Computerlinguistik annotiert werden. Somit standen 330 mit SCHUG annotierte Textdokumente für einen Versuch zur Verfügung. Für den Versuch wird das annotierte Korpus in OntoLT importiert, aus welchem dann Konzepte extrahiert werden. Zur Extraktion wurde die bereits definierte Mapping-Regel „*HeadNounToClass_ModToSubClass*“, verwendet, mit der Nomen, die statistisch als relevant eingestuft werden, zu einem Konzept und dessen Modifikatoren, die Wörter, Phrasen oder Sätze näher bestimmen (z.B. attributive Adjektive etc.), auf eine oder mehrere Unterklassen abgebildet werden [vgl. BUITELAAR ET AL. 2004:37].

Die extrahierten Konzepte werden dann aufgelistet und können nach Überprüfung durch den Benutzer zur Ontologie hinzugefügt werden. Um sich auf die wichtigen Konzepte konzentrieren zu können, die für die spezifische Domäne relevant sind, kann OntoLT eine statistische Relevanzbewertung der extrahierten Terme durchführen, die anschließend in die

Mapping-Regeln aufgenommen werden. Bei der Analyse werden die Frequenzen derselben Terme innerhalb des domänen-spezifischen Korpus mit der Frequenz der gleichen Terme in einem Referenzkorpus verglichen. Dieser Referenzkorpus ist allgemeiner gehalten als der domänen-spezifische, wodurch das Vorkommen von Wörtern innerhalb einer Domäne in Kontrast zu einem allgemeineren Gebrauch dieses Wortes gestellt wird. Als Referenzkorpus dient das Britisch National Corpus der Universität Oxford (BNC)[10]. Dieses Korpus stellt eine Momentaufnahme der englischen Sprache dar und soll einen Einblick über das Vorkommen von englischen Wörtern im heutigen Sprachgebrauch geben. Für diese statistische Analyse wurde die vorher bereits erwähnte „chi-square"-Analyse angewandt. Nach der Analyse werden die Ergebnisse aufgelistet, geordnet nach berechneter Relevanzpunktzahl. Nun kann der Benutzer wiederum einzelne Terme auswählen, die er für domänen-spezifisch hält. Die Regel wird also um weitere Bedingungen erweitert, nämlich um die ausgewählten Nomen. Nach Erweiterung der Regeln wurde diese wiederum auf das Korpus angewandt, so dass domänen-spezifische Konzeptkandidaten extrahiert werden. Dabei werden allerdings die Ergebnisse auf die in der Mapping-Regel enthaltenen Nomen beschränkt, welche dann manuell ausgewählt werden, um zu einer Ontologie hinzugefügt zu werden.

Für den Lernversuch soll die Mapping-Regel „*HeadNounToClass_ModifierToClass*" erweitert werden [vgl. BUITELAAR ET AL. 2004:38].

Das Ergebnis dieses Ansatzes ist in der beigefügten DVD 1 – Anlage 1.3 zu finden.

Im Vergleich zu TextToOnto steht bei OntoLT eine semi-automatische Vorgehensweise bei der Erstellung von Ontologien im Vordergrund. Somit ist OntoLT während der Erstellung einer Ontologie auf die Interaktion mit einem Experten angewiesen. BUITELAAR ET AL. (2004:8) beschreiben, dass künftig eine aktive Lernkomponente in OntoLT integriert werden könnte, die die Generierung von Operatoren und Regeln aufgrund eines Trainingsprozesses erlernt, indem frühere Spezifikationen des Experten gelernt werden.

[10] http://www.natcorp.ox.ac.uk

5.4 Probleme und eingeschlagene Richtungen

Bei der Erschließung der Teilkorpora und bei den Lernvorgängen traten diverse Probleme auf, auf die an dieser Stelle kurz eingegangen werden soll.

Bei HTML-Dateien, die von den von WILHELM (2004) erschlossenen Internetseiten heruntergeladen wurden traten Probleme nach der Konvertierung in reinen Text auf. Teilweise kam es vor, dass deutschsprachige Dateien enthalten waren oder Seiten auch nur serverseitige Fehlermeldungen (z.B. Fehler 404 etc.) enthielten. Da in der Datenbank auf Sites verwiesen wird, welche sowohl deutsch- als auch englischsprachig sind und diese beim Download berücksichtigt wurden, kann es sein, dass einige deutsche Dokumente im Korpus enthalten sind. Auch traten teilweise Fehler auf, wenn dynamische Webseiten mit Anawave WebSnake heruntergeladen werden sollten, so dass die Inhalte dieser nicht extrahiert bzw. konvertiert werden konnten. Alle erfassten Dateien wurden stichprobenhaft manuell überprüft und fehlerhafte Dokumente ausgesondert. Dabei wurden auch alle Dateien, die keinerlei Text enthielten, entfernt.

Auch bei dem Download der Dokumente von CiteSeer traten Probleme auf. Nachdem ein Großteil der Dokumente mit MyCrawler bereits erschlossen und konvertiert war, stellte sich heraus, dass die konvertierten PDF-Dateien erhebliche Fehler aufwiesen. In den Textdateien standen, nicht wie erwartet, der extrahierte Text, sondern nur Folgen unzusammenhängender Buchstaben und Sonderzeichen. Da aber fast 90% der PDF-Dokumente bei der Konvertierung diese Mängel aufwiesen, wurde ein anderer Weg versucht. Möglicherweise entstanden diese fehlerhaften Dateien nicht aufgrund einer fehlerhaften Anwendung beim Konvertieren. Es wurde neben der implementierten Konvertierungsmöglichkeit in MyCrawler und der Verwendung von PDF2Text auch noch versucht, einzelne Dokumente mit dem Adobe Acrobat Reader zu konvertieren. Diese Standardsoftware zur Betrachtung von PDF-Dokumenten liefert auch eine Option, mit der man PDF-Dokumente als Textdateien abspeichern kann. Auch unter Verwendung dieser Option blieb die Konvertierung weiterhin fehlerbehaftet. Aufgrund dieser Mängel wurden mehrere Dokumente manuell als PostScript-Dateien heruntergeladen und konvertiert. Die Vorgehensweise, erst PostScript-Dateien zu verwenden, diese dann in PDF-Dokumente zu konvertieren und daraus wiederum Textdateien zu generieren, erwies sich als Teillösung für die Konvertierungsproblematik. Nicht alle erschlossenen Dokumente konnten erfolgreich konvertiert werden. Da das gesamte Korpus

nun komplett erneut erschlossen werden sollte, wurde hierfür MyCrawler derart umprogrammiert und an die neue Situation angepasst, so dass nun ausschließlich PostScript-Dateien von CiteSeer berücksichtigt werden.

Des Weiteren trat bei der Arbeit mit CiteSeer ein Problem mit MyCrawler auf. Bei der Verwendung von MyCrawler kam es zu regelmäßigen Time-Outs aufgrund der schlechten Internetverbindung zu CiteSeer. Dies ist ein bekanntes Problem, da die Internetseiten von CiteSeer.com oftmals sehr lange laden bzw. nicht erreichbar sind, was wohl auf den CiteSeer-Server und hohen Traffic zurückzuführen ist. Diese schlechte Verbindung löste bei MyCrawler oft einen kompletten Abbruch aller laufenden Arbeitsvorgänge aus, so dass diese komplett neu gestartet werden mussten, da der Download nicht an der abgebrochenen Stelle fortgesetzt werden konnte. Aufgrund dessen wurde MyCrawler derart angepasst, dass nun abgebrochene Downloads an einer beliebigen Stelle fortgesetzt werden können.

Die Konvertierung der PDF-Dokumente mit dem in MyCrawler implementierten Modul funktioniert. Allerdings ist dazu zu sagen, dass bei einer Anzahl von mehr als 20 Dokumenten der Konvertierungsvorgang sehr lange dauern kann, da jede Konvertierung einen einzelnen Prozess startet und bei einem Kompletten Ordner diese Prozesse dann gleichzeitig ablaufen. Da bei der Vielzahl der im Korpus enthaltenen Dokumente dies zu lange dauert, wurde zusätzlich ein Programm namens PDF2Text eingesetzt, welches wesentlich schneller arbeitete.

Die Textdokumente, die man aus den PDF-Dokumenten erhält, enthalten den Text einer PDF-Datei. Allerdings kommt es vor, dass durch den textlichen Aufbau eines PDF-Dokuments Spalten fehlerhaft konvertiert werden, da nicht spalten- sondern zeilenweise bei der Konvertierung vorgegangen wird, wodurch zusammengehörige Textpassagen getrennt voneinander stehen. Wie bereits bei den konvertierten Dokumenten aus der Datenbank von WILHELM (2004) wurden beim CiteSeer-Korpus anschließend alle erfassten Dateien stichprobenhaft manuell überprüft und fehlerhafte Textdateien dabei ausgesondert.

Während der Durchführung der Lernversuche traten diverse Probleme auf, die teilweise den Ablauf der Experimente oder die gesamten Versuchsanordnungen beeinflussten. Die Versuche mit TextToOnto und dem Korpus mit den erschlossenen Internetsites konnte nicht erfolgreich durchgeführt werden. Dies könnte auf Konvertierungsfehler oder die sehr große Anzahl an Dokumenten pro Site zurückzuführen sein. Für alle weiteren Versuche wurden nur

die von CiteSeer erschlossenen Dokumente berücksichtigt. Bei den konvertierten PDF-Dokumenten gab es das Problem, dass aufgrund der Konvertierung in Textdateien mehrere Tokens ohne Leerzeichen aneinander gereiht im Text standen. Da sehr viele der konvertierten Dateien auch sehr groß waren und oftmals auch Literaturverzeichnisse, enthaltene Formeln oder Sonderzeichen fehlerhaft im Text dargestellt wurden, wurde eine repräsentative Menge des erschlossenen CiteSeer Korpus manuell nachbereitet. Dabei wurden nur die Titel der Veröffentlichungen und deren Abstracts in die Dokumente aufgenommen. Anschließend wurden Trennzeichen und fehlerhaft konvertierte Stellen manuell korrigiert. Da für den Bereich Computerlinguistik insgesamt nur 423 Dokumente zur Verfügung standen, wurden hieraus alle auf die gerade beschriebene Weise manuell nachbereitet. Damit aus all den für Information Retrieval berücksichtigten CiteSeer-Kategorien und -Suchbegriffen eine gleiche Anzahl an Dokumenten vorhanden war wie für Computerlinguistik, wurde pro Kategorie des Bereiches Information Retrieval jedes vierte Dokument ausgewählt und nachbereitet. Letztendlich standen somit für den Bereich Computerlinguistik 396 Dokumente und für den Bereich Information Retrieval 427 Dokumente zur Verfügung. Fehlerhafte Dokumente wurden dabei entfernt. Unter Verwendung der Abstracts konnten dann diverse Versuche mit TextToOnto erfolgreich durchgeführt werden. Somit war das Korpus kleiner als anfangs geplant.

Für OntoLT lag das Problem vor, dass die Dokumente mit SCHUG annotiert werden mussten. Da SCHUG derzeit nicht öffentlich zur Verfügung steht, wurde mit Hilfe von Paul Buitelaar und Alexander Schutz des DFKI Saarbrücken das Citeseer-Korpus für den Bereich Computerlinguistik annotiert, so dass nach Aussortierung fehlerhafter Dateien 330 annotierte Dokumente mit OntoLT verwendet werden konnten. Hierbei handelt es sich auch um vorher manuell nachbereitete Dokumente, da SCHUG Probleme mit Literaturverzeichnissen und Formeln hatte, so dass auch hier pro Datei der Titel und das Abstract einer Veröffentlichung verwendet wurden.

6 Darstellung und Evaluierung der Ergebnisse

In diesem Kapitel werden die Ergebnisse der Lernversuche präsentiert. Dabei werden zunächst Evaluierungsmöglichkeiten beim Ontology Learning vorgestellt und anschließend die Ergebnisse erläutert und evaluiert.

6.1 Ansätze zur Evaluierung

Prinzipiell ist zu sagen, dass das Ergebnis eines Lernvorgangs zur Erstellung einer Ontologie dann als korrekt oder gut einzuschätzen ist, wenn die maschinell gelernte Ontologie qualitativ mit einer manuell erstellten zu vergleichen ist. Die Bewertung von maschinell gelernten Ontologien ist diesbezüglich nicht trivial, da der Lernvorgang unüberwacht erfolgt. Dadurch, dass beim unüberwachten Lernen keine bereits bestehende Ontologie zum Trainieren verwendet wird, sondern eine Ontologie unüberwacht aus dem Korpus gelernt wird, muss anschließend überprüft werden, inwieweit ein Lernvorgang ein gutes oder ein schlechtes Ergebnis produziert hat.

Zur Durchführung einer Evaluierung, schlagen STAAB ET AL. (2003) zwei mögliche Ansätze für die Evaluierung der Ergebnisse vor: den praktischen Einsatz der Ontologie in einer Applikation und den Vergleich der Ontologie mit einer Referenzontologie, dem so genannten „Goldstandard“. Im praktischen Einsatz der Ontologie in einer Applikation werden Probleme und Unstimmigkeiten bei der Verwendung der Ontologie direkt augenscheinlich [vgl. STAAB ET AL. 2003]. Zum Beispiel kann anhand von Benutzertests festgestellt werden, inwieweit sich die Ontologie zum Browsing innerhalb der Bezugsdomäne eignet. Der Vergleich der erstellten Ontologie mit einem so genannten „Goldstandard“ ist vor allem hilfreich zur Aufdeckung und Interpretation von Unterschieden. Der Goldstandard stellt eine von Experten manuell erstellte und überprüfte Vergleichsontologie dar, welche die Konzepte einer spezifischen Domäne in einer korrekten und für die Experten gültigen Beziehung repräsentiert.

Bei der Verwendung einer Referenzontologie schlagen MAEDCHE & STAAB (2002) vor, die Ähnlichkeiten zwischen Ontologien zu messen, um daraufhin Aussagen über die erstellte Ontologie treffen zu können. In diesem Ansatz werden die lexikalischen und die konzeptualen Ebenen der Ontologien untersucht und darauf basierend Ähnlichkeitsmaße berechnet [vgl. MAEDCHE; STAAB 2002].

Auch DOAN ET AL. (2004:389) setzen bei der Überprüfung von Ontologien Ähnlichkeitsmaße ein. Dabei wird die untersucht, welche Auswirkungen Ähnlickeitsbeziehungen auf die Bedeutungen innerhalb einer Ontologie haben:

> *„Given two taxonomies and their associated data instances, for each node (i.e. concept) in one taxonomy, find the most similar node in the other taxonomy, for a pre-defined similarity measure."*

Bei den Ansätzen von DOAN ET AL. (2004) sowie MAEDCHE & STAAB (2002) muss jedoch ein Goldstandard vorhanden sein, der zum Vergleich herangezogen werden kann. Für die durchgeführten Versuche liegt es daher nahe, die Ontologie des virtuellen Bibliotheksregals MyShelf als Goldstandard einzusetzen, da diese manuell erstellte Konzepte für den Bereich Informationswissenschaft beinhaltet, unter anderem auch Computerlinguistik (Sprachtechnologie) und Information Retrieval (siehe Hanke-Klassen 4 und 9 im Anhang I). Eine Ontologie für MyShelf wurde im Rahmen des Projektseminars "Semantic Web und Ontologien" an der Universität Hildesheim im Wintersemester 2003/2004 basierend auf der Magisterarbeit von HANKE (2002) entwickelt und könnte hierfür eingesetzt werden. Diese soll im Folgenden als Hanke-Ontologie bezeichnet werden.

Zur Evaluierung der Lernergebnisse dieser Arbeit wurde eine Expertenevaluierung durchgeführt. Hierbei wurden die Ergebnisse unter domänen-spezifischen Aspekten untersucht und bewertet. Die taxonomische Struktur der gelernten Ontologien ist zum Beispiel für einen Browsingtest nicht geeignet, da die Konzepte sehr weit in die Tiefe gehen und Konzepte untereinander zu oft in keiner sinnvollen oder domänen-spezifischen Beziehung stehen. Für die vorgestellten Evaluierungstechniken, den praktischen Einsatz der Ontologien, den Vergleich mit einer Referenzontologie, liegt die Problematik folglich darin, dass die Ergebnisse sich strukturell zu stark von der Hanke-Ontologie unterscheiden. Dies verhindert folglich auch die Berechnung von Ähnlichkeitsmaßen. Somit wurde eine andere Möglichkeit gesucht, die Evaluierung durchzuführen. Auch weisen CIMIANO ET AL. (2004) darauf hin, dass sich allgemein bei der Evaluierung von Ontologien die Frage stellt,

„inwiefern automatisch gelernte Ontologien tatsächlich mit einem gewissen ‚Goldstandard' verglichen werden können". Gleichzeitig schlagen sie vor „alternativ[…] Menschen in die Evaluierung einzubeziehen, indem sie Vorschläge des Systems zu Relationen annehmen oder ablehnen".

SEIPEL & BAUMEISTER (2004) haben basierend auf GÓMEZ-PÉREZ (1999) einen Ansatz entwickelt, bei dem die taxonomische Struktur ohne den Vergleich mit einer Referenzontologie intellektuell auf Inkonsistenz, Unvollständigkeit und Redundanz überprüft werden soll. Zur Untersuchung der Inkonsistenz wird geprüft, ob es widersprüchliche Definitionen von einzelnen Konzepten gibt oder ob widersprüchliches Wissen aus den Definitionen abgeleitet wird [vgl. SEIPEL; BAUMEISTER 2004:53]. Zum einen spielt semantische Inkonsistenz der Konzepte eine Rolle, zum anderen ist die Akkuratheit semantischer Klassifikationen Untersuchungsgegenstand, d.h. inwiefern Unterkonzepte in die angemessene Oberklasse eingeordnet oder semantisch in eine falsche Beziehung gestellt wurden. Ontologien können unvollständig sein, wenn die is-a-Beziehungen zwischen den Konzepten nicht präzise definiert sind. Dies geschieht, wenn wichtige Konzepte ausgelassen oder nicht berücksichtigt werden. Auch wird die gelernte Ontologie auf redundante Konzepte oder is-a-Beziehungen hin überprüft [vgl. SEIPEL; BAUMEISTER 2004:54ff].

Aus diesem Grund wurde die Evaluierung stattdessen durchgeführt, indem manuell nach intuitiven Übereinstimmungen von gelernten Konzepten mit bekannten Konzepten der Domäne gesucht wurde. Dies ähnelt der Vorgehensweise, die SEIPEL & BAUMEISTER (2004) entwickelt haben

Die Ergebnisse werden nun daraufhin untersucht, ob einige gelernte Konzepte und taxonomische Strukturen für die jeweilige spezifische Domäne annähernd repräsentativ und semantische Zusammenhänge korrekt wiedergegeben sind. Dadurch, dass bei verwendeten Tools kein direkter Einblick in die Vorgehensweise möglich ist und somit auch einzelne Zwischenschritte während der Lernvorgänge zur Evaluierung nicht hinzugezogen werden können, stellen die Lernvorgänge selbst eine Art „Blackbox" dar. Da keine Einsicht in interne Ergebnisse und somit einzelne Lernschritte möglich ist, können Informationen folglich nur aus dem Vergleich unterschiedlicher Lernergebnisse unter kontrolliert veränderten Lernbedingungen gewonnen werden. Ebenso kann die Vorverarbeitung der Korpora durch den Standardprozessor nicht genauer untersucht werden, um herauszufinden, ob in diesem

Arbeitsschritt eventuell bereits Fehler entstehen und das Korpus somit fehlerhaft weitergegeben wird.

6.2 Ergebnisse der Lernversuche

Für die Versuche wurden Parameter auf zwei Ebenen variiert: die Korpora, aus welchen gelernt werden sollte, und die Lernverfahren selbst. Bei den Versuchen mit TextToOnto wurden Korpora der Domänen Computerlinguistik und Information Retrieval verwendet. Aufgrund erster Lernergebnisse wurde festgestellt, dass für weitere Lernversuche später nur noch mit den in den konvertierten PDF-Dokumenten enthaltenen Abstracts und Titel der Veröffentlichungen gearbeitet werden sollte.

Die verwendeten Lernverfahren bei TextToOnto waren der kombinations-basierte Ansatz und die formale Begriffsanalyse. Bei beiden Verfahren konnte die Anzahl der häufigsten Terme, die zum Lernen von Konzepten berücksichtigt werden sollten, manuell festgelegt werden (siehe Tabelle 4). Die Implementierung des kombinations-basierten Ansatz in TextToOnto bietet ferner weitere Möglichkeiten zur Variation, da beim Lernen optional Hearst-Pattern, Heuristiken und WordNet hinzugezogen werden können. Wie vorher schon beschrieben, konnten die erschlossenen Websites der Datenbank nicht erfolgreich gelernt werden. Auch konnten nicht alle CiteSeer-Dokumente des Bereiches Information Retrieval erfolgreich eingesetzt werden. Aus diesem Grund wurde eine repräsentative Menge an Dokumenten aus diesem Korpus verwendet, die je nur ein Viertel der Dokumente pro erschlossener CiteSeer-Kategorie beinhaltete.

6.2.1 Ergebnisse des kombinations-basierten Ansatzes

Allgemeine Tiefenstruktur

Der folgende Abschnitt beschreibt die Ergebnisse der Lernvorgänge mit TextToOnto. Die Versuchsanordnungen sind im Anhang VI aufgeführt. Die dazugehörigen Dateien befinden sich auf der beigelegten DVD 1 - Anlage 1.2.

Bei den Ergebnissen des kombinations-basierten Ansatzes fällt auf, dass sehr viele der berücksichtigten Terme jeweils nur zu einem einzigen Konzept in Beziehung stehen. Dies führt dazu, dass die hierarchische Struktur sehr weit in die Tiefe wächst und nicht in die Breite geht (siehe Abbildung 17).

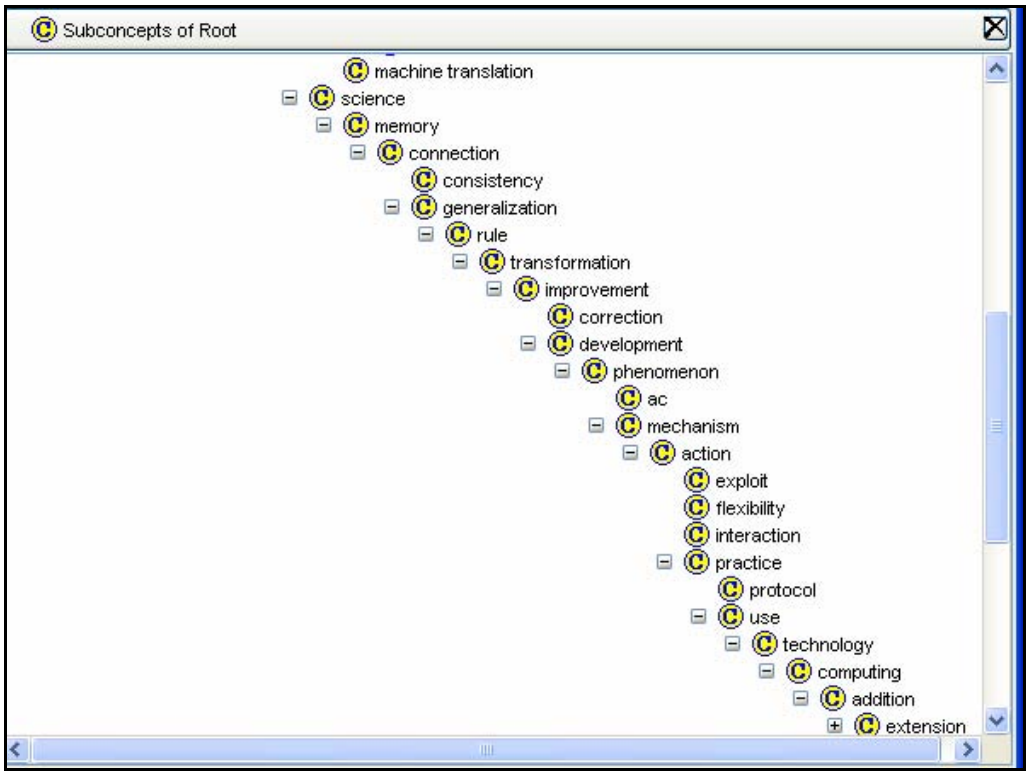


Abbildung 17: Tiefenstruktur in Modell 07.

Auffallend hierbei ist, dass es dennoch Konzepte gibt, die mehr als nur ein Unterkonzept besitzen. Diese befinden sich jedoch nicht auf den oberen Ebenen der Taxonomie, sondern liegen einzeln und sehr verstreut in der Tiefe. Dies stellt insofern ein Problem dar, als dass diese einzelnen untergeordneten Konzepte in der Abfolge der Hierarchieebenen als semantisch nicht korrekt zueinander in Beziehung stehend angesehen werden können, da die

in Kapitel 2.3.1 beschriebene Transitivität der Konzepte verletzt wird. Im Vergleich dazu besitzt eine domänen-spezifische Ontologie wie die Hanke-Ontologie ein Maximum von vier Hierarchieebenen. Die Hanke-Ontologie soll an dieser Stelle nicht als Goldstandard angesehen werden, sondern nur einen annähernden Bezugspunkt für den strukturellen Aufbau einer verwendbaren Ontologie darstellen. Die Struktur der oben beschriebenen gelernten Ontologie macht diese unüberschaubar und erschwert das Erkennen von semantischen Beziehungen der Konzepte untereinander bzw. stellt diese nicht korrekt dar.

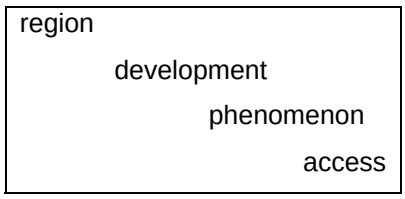


Abbildung 18: Beispiel 1 aus Modell 02.

Dies wird zum Beispiel in Abbildung 18 deutlich. Diese zeigt Konzepte aus dem Modell 02, die in einer is-a-Beziehung stehen. Die verschiedenen Ebenen werden durch unterschiedlichen Einschub dargestellt. Das Konzept *development* ist kein sinnvolles Unterkonzept von *region*, ebensowenig wie *phenomenon* als semantisches Unterkonzept von *development* aufzufassen ist. Die Struktur der gelernten Beziehungen lässt sich auch nicht erklären, wenn man eine Verbindung der Konzepte über ihre synonymen oder polysemen Bedeutungen ins Auge fasst oder indem man die Bedeutungen großzügig auslegt. Derartig inkonsistente Zuordnungen setzen sich durch die Lernergebnisse fort.

Bildung struktureller Inseln

Verfolgt man diese in die Tiefe gehenden Äste weiter, so stößt man in unregelmäßigen Abständen auf „Inseln“. Damit sind Gruppierungen von Konzepten gemeint, die mehrere Konzepte beinhalten und eher in die Breite gehen, wobei diese Gruppen vereinzelt in tieferen Ebenen vorkommen. Wegen der erwähnten Tiefenstruktur der Ergebnisse ermöglichen diese Inseln erst eine inhaltliche Betrachtung, da die Tiefe die Untersuchung vereinzelter Konzepte aus strukturellen Gründen nicht zulässt. Aufgrund der Ergebnisse kann nicht gesagt werden, wann diese Inseln auftreten bzw. welche Faktoren darauf Einfluss haben. Auch hier ist nicht zwingend ein semantischer Bezug zu erkennen, jedoch kommen in diesen größeren Gruppierungen ab und zu sinnvolle Zusammenhänge der Konzepte zum Vorschein. Dabei werden Begriffe in eine an sich sehr schlüssige Beziehung zueinander gestellt, allerdings

besteht oft kein semantischer Bezug zur Domäne, obwohl das Lernkorpus domänen-spezifisch ist. In Abbildung 19 wird dies deutlich. Das Konzept time beinhaltet die Konzepte *future*, *past*, *present*, *second* und *while*, was alltagssprachlich eine sinnvolle Zusammenstellung ergibt. Die Verbindung dieser Konzepte unter *time* ist zwar sinnvoll und korrekt, jedoch nur für einen allgemeinen Bezugsrahmen. Innerhalb der Domäne Information Retrieval, aus der das Korpus bei diesem Lernvorgang stammt, hat das Konzept time keinerlei Aussagekraft.

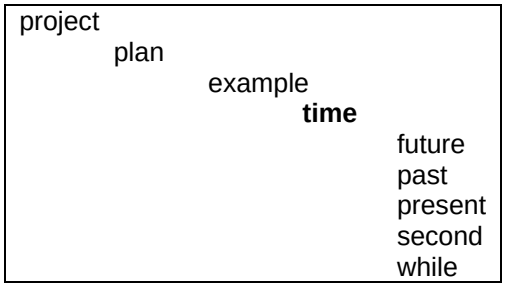


Abbildung 19: Beispiel 2 aus Modell 15.

Allerdings kann man auch sehr oft feststellen, dass auch dort, wo Inseln auftreten, diese selbst eine Unterkategorie zu einem Konzept bilden, mit welchem sie semantisch nicht in Verbindung stehen, wie hier am Beispiel aus Modell 15 in Abbildung 19 an dem Konzept time mit dem Oberkonzept *example* deutlich wird. „Zeit“ ist kein „Beispiel“, sondern vielmehr ein Maß. Auch ist anzumerken, dass semantisch sinnvolle Konzepte aufgrund der Tiefe nicht in der gesamten Abfolge der Hierarchie gesehen werden können. Von dem Beispiel aus Tabelle 2 kann man also nicht sagen, dass *future* eine fortgesetzte Spezifizierung von *example* ist. Taxonomien beinhalten auf den höheren Ebenen allgemeinere Konzepte und werden spezifischer, je weiter man in die Tiefe vordringt. Durch die extreme Verzweigung in die Tiefe ist innerhalb der Ergebnisse nicht mehr von spezifischeren Konzepten auszugehen. Dies zeigt sich in Beispiel 2 auch darin, dass viele Unterkonzepte (*example*) semantisch keinerlei Bezug zu ihrem Oberkonzept (*plan*) oder dessen Oberkonzept (*project*) haben. Auch wenn das Oberkonzept einer Insel semantisch zutreffend ist, so hat es meistens eine sehr allgemein gehaltene Bedeutung, wie am Beispiel 3 in Abbildung 20 ersichtlich ist. Dort steht das Konzept *person* sehr allgemein unter *class*. Trotz einer sinnvollen Klassifikation von *candidate*, *expert*, *learner*, *researcher* und *user* als Personenkonzepte enthält das Beispiel auch fehlerhaft zugeordnete Konzepte (*hierarchy* und *power*) unter demselben Konzept.

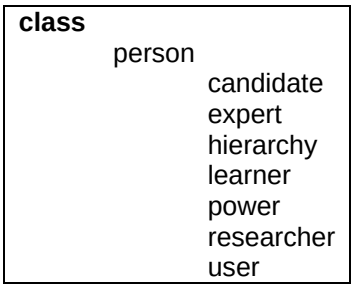


Abbildung 20: Beispiel 3 aus Modell 07.

Einfluss der Anzahl der häufigsten Terme

Eine Analyse der Lernergebnisse, bei denen die Anzahl der als Konzeptkandidaten zu berücksichtigenden Terme variiert wurde, zeigte, dass diese Zahl einen Einfluss auf die Tiefe der Hierarchie hat. Je mehr Terme zum Lernen berücksichtigt werden, desto mehr Konzepte werden in Beziehung zueinander gestellt und untersucht. Weniger Terme führen zu präziseren Kategorien und weniger tiefen Ontologiestrukturen. Dadurch dass bei dem Lernverfahren die häufigsten Terme des Korpus verwendet werden, bedeutet dies allgemein erst mal nicht, dass folglich ein besseres Ergebnis bezüglich der semantischen Aussagekraft erzielt wird. Allerdings lässt sich feststellen, dass mit steigender Anzahl Konzeptkandidaten auch mehr domänen-spezifische Konzepte in der Ontologie enthalten sind - ohne dass dies allerdings einen feststellbaren Einfluss auf die Qualität der semantischen Zusammenhänge dieser Konzepte hätte.

Um den Einfluss der Anzahl der häufigsten Terme beim Lernen darzustellen, wurde anhand von intuitiv ausgewählten Beispielkonzepten überprüft, wie viele als relevant eingestufte Konzepte in den Ergebnissen enthalten sind. Da wie bereits erwähnt keine Referenzontologie zum Vergleich hinzugezogen werden konnte, wurde die Evaluierung intuitiv durchgeführt. Diese orientierte sich zum einen an Konzepten, die in der Hanke-Klassifikation für diese Teilbereiche enthalten sind und zum anderen daran, welche Konzepte intuitiv als relevant empfunden wurden, so dass diese unbedingt für die jeweiligen Teilbereiche in der Ontologie enthalten sein sollten.

Für die Bereiche Computerlinguistik und Information Retrieval wurde eine Auswahl an Beispielkonzepten festgelegt, die als relevant hinsichtlich der Domänen eingestuft wurden (siehe Tabelle 5). Hierbei wurden Konzepte nach folgenden Kriterien ausgewählt: ein Teil der Beispielkonzepte Konzepte wurde anhand der Klassen der Hanke-Ontologie für die Bereiche

Computerlinguistik und Information Retrieval ausgewählt (siehe Hanke-Klassen 4 und 9 Anhang I). Die anderen Beispielkonzepte wurden für die beiden Fachbereiche intuitiv festgelegt. Dabei wurde berücksichtigt, dass auch Konzepte ausgewählt wurden, die bei den ersten Untersuchungen der Ergebnisontologien als domänen-spezifisch eingestuft werden konnten. Somit wurde eine Untersuchungsgrundlage geschaffen, bei der zum einen als relevant bewertete Konzepte verwendet wurden, und bei der zum anderen von den Konzepten teilweise bekannt war, dass diese auch in den Ontologien enthalten waren.

Computerlinguistik	**Information Retrieval**
annotation	classification
artificial intelligence / ai	data mining
computational linguistics	digital libraries
language technology	extraction
linguistics	filtering
machine translation	information retrieval
natural language processing / nlp	knowledge discovery
parsing / parser	meta search
semantics	retrieval system
speech recognition	search engine
syntax	recall
tagger	precision
word sense disambiguation	software agents / agents
	information system
	relevance feedback

Tabelle 5: Intuitiv festgelegte Konzepte für die Bereiche CL und IR.

Aufgrund dieser Referenzkonzepte wurde dann mittels einer Suchfunktion in TextToOnto überprüft, welche hiervon in den Ergebnissen enthalten waren. Aufgrund des Vorkommens relevanter Konzepte in den einzelnen Ontologien wurden dann Werte für den Recall folgendermaßen berechnet:

$$\frac{\text{Anzahl der relevanten Konzepte, die in der Ontologie enthalten sind}}{\text{Alle als relevant festgelegten Konzepte}}$$

Die ausführliche Berechnung der Recall-Werte ist dem Anhang VIII zu entnehmen. Die Berechnung des Recall ergibt in Bezug auf die Anzahl der bei den Lernvorgängen (mit

Hearst-Pattern, Heuristiken und WordNet) berücksichtigten häufigsten Terme folgendes Ergebnis (siehe Tabelle 6):

	Anzahl Terme 50	Anzahl Terme 150	Anzahl Terme 500
CL Volltext	-	0,077	0,615
CL Abstracts	0,154	0,538	**0,846**
IR Volltext	0,133	0,200	0,267
IR Abstracts	-	0,400	0,733
Alle abstracts	0,036	0,179	0,643

Tabelle 6: Recall für verschiedene Parameter.

Tabelle 6 zeigt, dass die Recall-Werte größer werden, je höher die Anzahl der Terme gesetzt wird. Dies bedeutet, dass je mehr Terme als potentielle Konzeptkandidaten bei den Lernvorgängen berücksichtigt werden sollen, desto mehr der als relevant eingestuften Beispielkonzepte sind in der gelernten Ontologie enthalten. Somit kann für weitere Lernversuche empfohlen werden, die Anzahl der häufigsten Terme als Konzeptkandidaten zu erhöhen. Die höchste Trefferquote für den Bereich Computerlinguistik und für Information Retrieval wurde jeweils unter Verwendung der Abstract-Korpora erzielt.

Einfluss der verschiedenen Korpora

Die Verwendung eines Volltextkorpus im Vergleich zur Verwendung eines Korpus, der nur aus Abstracts besteht, zeigt, dass das Ergebnis keine großen Unterschiede aufweist. Die Verwendung von Volltexten führte zu mehreren kleineren Inseln, allerdings bleiben die Ergebnisse weiterhin domänen-unspezifisch.

Generell lässt sich beobachten, dass die Konzepte in einem sehr allgemeinen und domänen-unspezifischen Kontext gelernt wurden, obwohl die Korpora sich auf spezielle Domänen - nämlich Computerlinguistik, Information Retrieval oder beide - bezogen. Wenn man die Ergebnisse einfach durchsucht, lässt sich aufgrund der vorhandenen Konzepte nicht zwingend auf die Zugehörigkeit zu einer speziellen Domäne schließen. Das Konzept *document* in Abbildung 21 aus dem Modell 12 beinhaltet das allgemeinere Konzept *process*.

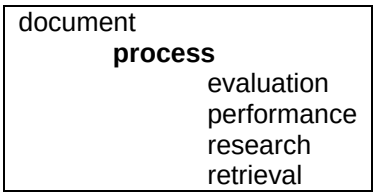


Abbildung 21: Beispiel 4 aus Modell 12.

Die beiden stehen semantisch nicht wirklich in einer is-a-Beziehung. Jedoch stellen die Unterkonzepte *evaluation*, *performance*, *research* und *retrieval* Verarbeitungsmöglichkeiten und –prozesse für Dokumente dar. Semantisch gesehen sind diese Beziehungen sinnvoll, allerdings nur in einem allgemeineren Kontext.

Einfluss der Hearst-Pattern, der Heuristik und von WordNet

Bei der Verwendung des Ansatzes ohne Hearst-Pattern (Modell 24) konnte man sehen, dass die hierarchische Struktur wiederum sehr in die Tiefe ausschweift. Dasselbe Ergebnis entsteht auch bei den Lernvorgängen ohne Heuristiken (Modell 25). Im Hinblick auf die Domäne des Korpus sind die Beziehungen zwischen den Konzepten wiederum nur sehr allgemein gehalten. Da WordNet bei beiden Vorgängen verwendet wurde, zeigt sich anhand des folgenden Beispiels, dass durch WordNet die hierarchische Struktur bezüglich der Tiefe beeinflusst wird. In Modell 23 wird der kombinierte Ansatz ohne die Einbeziehung von WordNet angewandt. Im Vergleich zur parallelen Versuchsanordnung in Modell 20 zeigt sich deutlich, dass ohne WordNet maximal zwei Hierarchieebenen entstanden sind, wohingegen in Modell 20 die Hierarchie sehr in die Tiefe geht. Auffallend hierbei ist, dass die Unterkonzepte jeweils nur aus den Mehrworttermen zu den Oberkonzepten bestehen, was man Abbildung 22 entnehmen kann. Es werden also keine weiteren semantischen Bezugsmöglichkeiten verwendet.

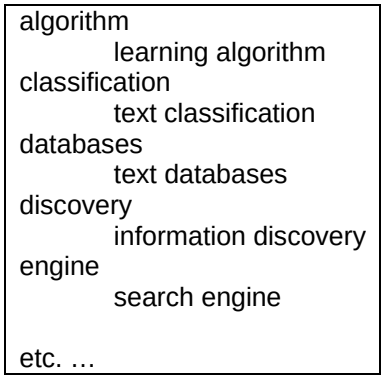


Abbildung 22: Modell 23 ohne WordNet.

Ohne WordNet können aufgrund fehlender semantischer Informationen nur Mehrwortterme als Unterkonzepte gelernt werden.

WordNet beeinflusst nicht nur die Tiefe der Taxonomie, sondern auch die Art, in welcher Terme miteinander in Beziehung gestellt werden. Dies zeigt sich in deutlich unterschiedlichen Recall-Werten für die gleiche Versuchsanordnung einmal mit und einmal ohne WordNet. Der Recall-Wert gibt Auskunft darüber, wie viele intuitiv als relevant eingestufte Konzepte in der Ontologie enthalten sind. Je höher der Wert ist, desto mehr relevante Terme sind enthalten. Überprüft man den Recall der Versuchsanordnung von Modell 23 – also die Anordnung ohne WordNet -, so ergibt sich ein Recall-Wert von 0,286. Für die gleiche Versuchsanordnung, jedoch unter Einbeziehung von WordNet (Modell 20), ergibt sich ein Recall von 0,643. Das bedeutet, dass unter Verwendung der gleichen Ausgangsbasis (Lernalgorithmus, Korpus) Terme unter Verwendung von WordNet offenbar anders in Beziehung gestellt werden, so dass dadurch mehr relevante Konzepte entstehen. Dies zeigt sich auch an dem oben gezeigten Beispiel des Modells 23, bei dem die Konzepte ohne die Berücksichtigung weiteren Quellen semantischer Information in eine sehr einfache Beziehung zueinander gestellt werden. Daraus lässt sich schließen, dass durch WordNet Terme besser in Beziehung gestellt werden, weil mehr domänen-spezifische Konzepte entstehen. Bei der Verwendung von WordNet wird die vorhandene Hyponymie-Struktur ausgenutzt, so dass Terme, die auf eine is-a-Beziehung untersucht werden, mit den Bedeutungen der Vergleichsterme, die aus WordNet bezogen werden, in Beziehung gesetzt werden. Hierdurch werden mehrere Bedeutungen aus WordNet für die Terme verwendet, um mehrere Vergleichskombinationen zur Überprüfung der Hyponymie auszuprobieren [vgl. CIMIANO ET AL. 2004]. Werden mehrere Bedeutungen hinzugezogen, so werden zwangsläufig auch die übergeordneten allgemeineren Bedeutungen verwendet. Dies wird in Abbildung 23 deutlich, die die Hyponymie-Struktur für *user* und *researcher* in WordNet zeigt. In Modell 07 stehen *researcher* und *user* unter dem Konzept *person* (siehe Abbildung 20). Ihre Position innerhalb der Hyponymie-Struktur in WordNet ist Abbildung 23 zu entnehmen. Wie man sieht, haben *user* und *researcher* die Oberkategorien von *object* bis *person* gemeinsam und fallen dann in verschiedene spezifischere Kategorien. Da nun allgemein bei den Vergleichspaaren mehrere Bedeutungen (Synonyme und Quasi-Synonyme) berücksichtigt werden, werden somit auch diese sehr generellen übergeordneten Bedeutungen wie *person* oder *object* hinzugezogen.

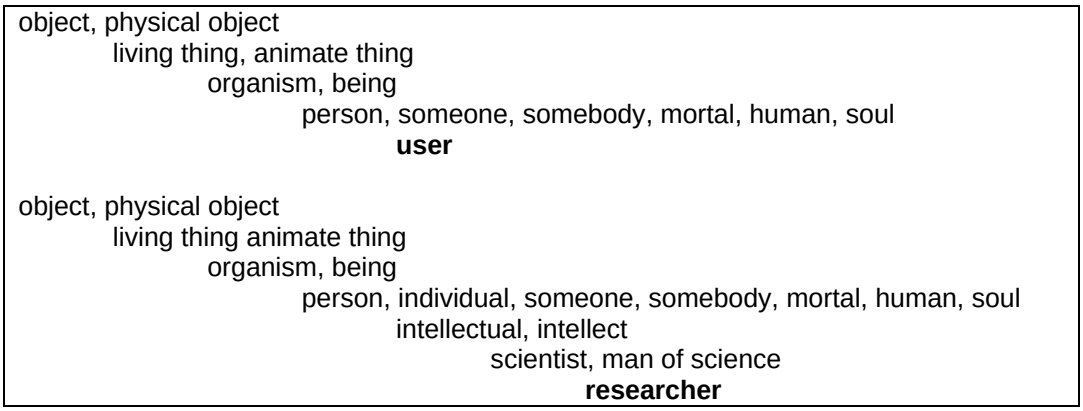


Abbildung 23: Hyponymie-Struktur aus WordNet für *user* und *researcher*.

Da diese aufgrund ihres hohen Allgemeinheitsgrades sehr vielen Konzepten zuzuordnen sind, werden sie auch öfter in der Ontologie berücksichtigt. Dadurch ergibt sich eine sehr starke allgemeine Ausrichtung der Konzepte zueinander, weil in den Vergleichspaaren natürlich die allgemeinen Klassen öfter vorkommen als spezifische. Zum einen liefern diese Vergleichsterme dann die Hyponymie-Struktur, und zum anderen würde dies erklären, warum die Konzepte sehr allgemein gehalten sind. Auch würde somit deutlich werden, dass nur sehr wenige Konzepte spezifisch auf eine spezielle Anwendungsdomäne zugeschnitten sind, da WordNet eine domänen-unspezifische Quelle ist und für die Anwendungsdomäne Computerlinguistik oder Information Retrieval keine spezifischen Konzepte enthält. So werden Terme aus den Domänenkorpora mit Termen aus WordNet in einem allgemeinen Kontext in Beziehung gestellt (siehe Abbildung 20 - Modell 07). Dadurch erhalten die Konzepte eine Abhängigkeitsstruktur, wenn auch eine sehr allgemeine, da WordNet keine domänen-spezifische Quelle ist.

Dies würde bedeuten, dass die Strukturinformation aus WordNet für den Lernvorgang eine wesentliche Rolle spielt. Da die semantischen Beziehungen innerhalb einer domänen-spezifischen Ontologie aber die Wissensstruktur der jeweiligen Domäne widerspiegeln sollten, sind die auf diese Weise entstandenen Ontologien jedoch kaum brauchbar.

6.2.2 Ergebnisse der formalen Begriffsanalyse

Die formale Begriffsanalyse bietet für den Lernvorgang zwei Optionen: die Berücksichtigung aller Verben im Korpus oder die Berücksichtig der lexikografischen Klassen von WordNet. Unter Verwendung der lexikografischen Klassen von WordNet konnte jedoch kein verwertbares Ergebnis gelernt werden, da die meisten Konzepte ohne Bezeichnungen, also ohne Namen, in der Hierarchie standen, so dass keine Zusammenhänge der Konzepte ersichtlich waren.

Unter Berücksichtigung aller Verben im Korpus enthielten die meisten Konzepte der Ergebnisse nur ein Objekt. Anders als beim kombinations-basierten Ansatz stehen hier für die Bezeichnungen der Kategorien Verben, die bestimmte Konzepte beschreiben sollen. Somit ist der Aufbau der Ontologie anders. Die Konzepte erstrecken sich auf maximal drei Ebenen und gehen nicht so sehr in die Tiefe wie beim kombinations-basierten Ansatz. Die gelernten Konzepte zeigen, dass die Beziehungen der Konzepte zu den beschreibenden Verben in den Konzeptklassen sehr oft nur in einem sehr allgemeinen Kontext stehen. Auch wurden nur sehr wenige als relevant eingestufte Konzepte in die Ontologie aufgenommen. Die Ontologie beinhaltet fast ausschließlich allgemeine Konzepte.

Konzepte	Beschreibende Verben
text	translate, classifying, tailoring, scan, comprise, supertagging, modelling, issuing, confuse
information	collecting, communicate, rapidlychanging, gaining, indicating, volunteer, merge, convey, extracting, retrieving, clarify, communicated, encode, carry, transferring, yield, disseminating, gather, preserve, gain, exchanging, enter, changing

Tabelle 7: Beispielergebnisse nach formaler Begriffsanalyse.

Tabelle 7 soll veranschaulichen, wie Konzepte durch Verben semantisch sehr ausführlich beschreiben werden. Dadurch, dass die verwendeten Verben die Verwendungsmöglichkeit der Konzepte darstellen, wird deutlich, in welchem semantischen Kontext diese auftreten können. Allerdings ist die Struktur der gelernten Ontologie für den Anwendungsfall eher schlecht geeignet, da zum Beispiel beim Browsing eine derartige Struktur eher umständlich in der Handhabung ist. Die ungewohnte Darstellung von Beziehungen ist so gesehen nicht sehr benutzerfreundlich, auch wenn diese Methode einen Weg darstellt, Konzepte anschaulich und

semantisch eindeutig zu beschreiben. Somit eignen sich diese Ergebnisse nicht zur Weiterverwendung bzw. für den Anwendungsfall Browsing.

6.2.3 Ergebnisse von OntoLT

OntoLT legt den Schwerpunkt auf semi-automatische Generierung. Somit werden Konzepte nach der chi-square-Analyse aus einem annotierten Korpus extrahiert und dem Benutzer zur Auswahl präsentiert, welche dann nach Akzeptanz von Seiten des Benutzers in die Mapping-Regel aufgenommen werden. Das Ergebnis der chi-square-Analyse liefert eine Reihe von bewerteten Termen. Diese Punktzahl ergibt sich aus dem Vergleich der Frequenz der Terme im Korpus mit der Frequenz der Terme im British National Corpus über die bereits erwähnte chi-square-Analyse.

Vor der Durchführung der chi-square-Analyse wurden 979 Nomen als Konzeptkandidaten vorgeschlagen. Nach der Analyse kamen davon 413 Terme mit der höchsten Punktzahl in die engere Auswahl (siehe DVD 1 - Anlage 1.4). Die Terme werden also aufgrund ihrer Frequenz im Korpus als auch aufgrund ihrer allgemeinen Benutzung in der englischen Sprache bewertet. Auch hier wurde auf die oben beschriebene Art der Recall berechnet, um herauszufinden, inwieweit relevante Konzepte von der chi-square-Analyse als relevant vorgeschlagen werden. Die Berechnung erfolgt parallel zu oben beschriebenen Recall-Werten. Für den Einsatz von 330 mit SCHUG annotierten Dokumenten ergibt sich ein Recall für die als relevant eingestuften Konzepte nach der chi-square-Analyse von 0,615. Knapp 62% der intuitiv als relevant festgelegten Konzepte stehen nach der Analyse in der Auflistung der Terme.

Dies zeigt, dass die Begriffe durchaus domänen-spezifische Konzepte enthalten, jedoch bei der Relevanzbewertung sehr viele allgemeine Begriffe eine hohe Punktzahl erreicht haben, als die intuitiv festgelegten spezifischen Konzepte aus der Recall-Berechnung (siehe Anhang VIII). Ab hier obliegt es dem Benutzer, die Terme nach eigenem Ermessen auszuwählen, so dass die Mapping-Regel um diese erweitert wird. Für künftige Extraktionsverfahren lässt sich sagen, dass die relevanten Begriffe in der Mapping-Regel auf ein anderes Korpus der selben Domäne angewendet werden kann, so dass der Auswahlprozess in gewissem Maße

automatisiert und dadurch der manuelle Aufwand reduziert wird. Die semantischen Beziehungen in OntoLT werden über den Benutzer festgelegt, OntoLT unterstützt lediglich bei der Vorauswahl potentieller Konzepte. Hat man nach der Analyse Konzepte ausgewählt und die Mapping-Regel um diese erweitert, werden bei einem erneuten Extraktionsvorgang aufgrund der aktualisierten Mapping-Regel aus dem Gesamtkorpus nur noch die in der Regel enthaltenen Terme extrahiert. Auffallend ist, dass hierbei Mehrwortterme, die einen dieser Terme enthalten, in der Ontologie als Unterkonzept dargestellt werden. Beispielsweise befindet sich unter dem Konzept *annotation* in der Ontologie Konzepte wie *annotation_manual*, *annotation_automatic*. Somit gibt es maximal zwei Hierarchieebenen, wobei die zweite Ebene immer die Mehrwortterme als Unterkonzepte beinhaltet. Dies kann darauf zurückzuführen sein, dass bei der Mapping-Regel unter anderem die Modifikatoren zu einem Nomen berücksichtigt werden, die dann auf eine oder mehrere Unterklassen abgebildet werden [vgl. BUITELAAR ET AL. 2004:37]. Nach der statistischen Analyse wurden aus der Liste der Konzepte (siehe DVD 1 - Anlage 1.3) die Oberklassen aus der Ontologie durch den Benutzer hinzugefügt. Bei der anschließenden Anwendung der erweiterten Regel auf das Korpus wurden die Oberklassen der Ontologie im Anhang VII vorgeschlagen, die dann zu der Ontologie hinzugefügt wurden.

OntoLT bietet durch die semi-automatische Vorgehensweise keine wirkliche automatische Erstellung der Konzepte aus dem Korpus in dem Sinne, dass das System automatisch Begriffe auswählt und deren semantische Beziehung definiert. Dies erfolgt komplett durch den Benutzer, der durch seine Expertise in der Domäne die Begriffe auswählen kann. Die Auswahl der Terme bleibt somit im Aufgabenbereich des Benutzers und wird durch OntoLT unterstützt. Das Lernergebnis von OntoLT zeigt auch, dass die so erstellte Ontologie weiterhin verfeinert und nachbereitet werden muss, um zum Beispiel für einen Browsingzugriff verwendet werden zu können.

6.3 Fazit

Allgemein wird an den Ergebnissen deutlich, dass diese ohne Überprüfung durch den Experten nicht eingesetzt werden können. Weiterhin gibt es verschiedene Ansätze, die eine Vorgehensweise und Evaluierungskriterien vorschlagen, jedoch hat man an den

Lernergebnissen gesehen, dass diese Ansätze hier in der vorgesehenen Art nicht angewandt werden konnten. Somit könnte die Weiterentwicklung und Verfeinerung von Evaluierungskriterien dazu beitragen, Ontologien besser und eindeutiger bewerten zu können. Dabei stehen jedoch die Bedürfnisse der Benutzer, sowie die Orientierung am gegebenen Anwendungsbereich im Mittelpunkt.

Die Versuche zum maschinellen Lernen von Ontologien haben verschiedene Lernergebnisse hervorgebracht. In den Ergebnissen wurde deutlich, dass bei der Verwendung einer hohen Anzahl an häufigsten Termen als Konzeptkandidaten bei den Lernversuchen mehr relevante Konzepte enthalten sind. Somit empfiehlt es sich, viele Terme für weitere Lernversuche zu benutzen. Bei der Verwendung des kombinations-basierten Ansatzes in TextToOnto haben die Ergebnisse gezeigt, dass diese ohne Nachbereitung durch einen Experten nicht verwendet werden können. Die Ergebnisse zeigen, dass nur ein kleiner Teil der gelernten Konzepte in einem sinnvollen Zusammenhang stehen, jedoch semantisch mit der Domäne Computerlinguistik oder Information Retrieval nicht in Beziehung stehen. Die Konzepte stellen zwar teilweise richtige Beziehungen dar, allerdings sind diese sehr allgemein gehalten. Auch konnte festgestellt werden, dass das Wissen über die Konzepte hierbei aus WordNet stammt, was wiederum den Allgemeinheitsgrad der Konzeptbeziehungen beeinflusst. Auch die Tiefenstruktur der Ergebnisse unter Verwendung von WordNet zeigt, dass bei diesem Ansatz eine bessere Balance gefunden werden müsste, damit die Hierarchie nicht zu stark in die Tiefe wächst. Dadurch, dass die hierarchische Struktur bei den Ergebnissen sehr weit in die Tiefe geht, könnte der Einsatz von Pruning-Verfahren hilfreich sein, um die oben beschriebenen „Inseln“ zu lokalisieren und auf höheren Ebenen zugänglich zu machen. Dabei müssten die Inseln herausgefiltert werden und von den überflüssigen Verzweigungen getrennt werden.

Die Verwendung der formalen Begriffsanalyse zeigte, dass die hierarchische Struktur zur Verwendung der Ontologie für einen Benutzer nicht geeignet ist. Jedoch stellt dieser Ansatz eine gute Möglichkeit dar, um Konzepte ausführlich über die Verben, die mit den Konzepten im Korpus in Zusammenhang stehen, zu beschreiben. Somit werden Konzepte semantisch detaillierter beschrieben, letztendlich obliegt es jedoch wiederum dem Experten, dieses generierte Wissen über die Konzepte sinnvoll in die Ontologie zu integrieren.

OntoLT liefert einen semi-automatischen Ansatz zur Ontologieerstellung. Hierbei unterstützt der Ansatz die Auswahl relevanter Terme, die letztlich durch den Experten aufgrund dessen

Domänenwissens ausgewählt und zur Ontologie hinzugefügt werden. Wie das Ergebnis unter Verwendung von OntoLT gezeigt hat, enthält die statistische Analyse viele der als relevant erachteten Konzepte. Auch wurde durch die Anwendung der statistischen Analyse die Auswahl relevanter Konzepte unterstützt. Allerdings kommt das Wissen über die Domäne oder über die Beziehungen relevanter Konzepte zueinander vom Experten selbst. Somit wird maschinell der Erstellungsvorgang der Ontologie unterstützt. Aus semantischer Sicht wird aber kein weiteres Wissen über die Domäne miteinbezogen. Durch die semi-automatische Erweiterung der Mapping-Regeln können diese für andere Anwendungsfälle auf andere Korpora der Domäne benutzt werden, so dass domänen-spezifisches Wissen über den Experten in Form dieser erweiterten Regeln weitergegeben werden kann, was an sich den manuellen Aufwand verringert.

In den Versuchen wurde deutlich, dass eine maschinell gelernte Ontologie nicht die Qualität einer manuell erstellten Ontologie besitzt. Zwar gibt es Ansätze, die bessere Ergebnisse erzielen als andere, jedoch bleibt das Eingreifen des Menschen unausweichlich, zumal Wissen nicht von der Maschine generiert wird. Da dies das Ziel bei maschinellen Lernverfahren darstellt, müssen die existenten Verfahren weiterentwickelt werden. Allerdings hat man gesehen, dass in allen Fällen eine Nachbereitung durch Experten erforderlich ist. Somit können rein maschinelle Lernverfahren ohne Interaktion eines Experten nicht zur Erstellung von Ontologien verwendet werden. Vielmehr stellt die semi-automatische Erstellung von Ontologien einen Ausgangspunkt dar, der die Erstellung von Ontologien sinnvoll unterstützt. Die Interaktion mit dem Menschen verspricht dabei die Übertragung des Domänenwissens in eine Ontologie. Die semi-automatische Vorgehensweise von OntoLT stellt demnach eine Alternative zu den vollautomatischen Verfahren des TaxoBuilders von TextToOnto dar, da hier die Eingliederung von Domänenwissen in die Ontologie über den Experten gewährleistet ist. OntoLT führt bei der Erstellung von Ontologien wichtige Schritte automatisch aus, wie z.B. die Extraktion von Konzepten in eine Auswahlliste oder die Durchführung der statistischen Analyse, die somit eine Unterstützung für den Experten beim gesamten Erstellungsprozess darstellen.

Auch muss an dieser Stelle darauf hingewiesen werden, dass Ontologien nach STAAB & STUDER (2004:VII) auf einem sozialen Prozess beruhen, der auf einer Einigung unter einer Gruppe von Leuten bezüglich der Wahl von Konzepten und Relationen basiert, die in die Ontologie integriert werden sollen. Somit steht neben der Wahl maschineller Lernverfahren

auch die Zusammenarbeit innerhalb der Expertengruppe untereinander im Mittelpunkt, um domänen-spezifische Ontologien in einem gemeinsamen Konsens zu erstellen.

In Bezug auf MyShelf kommt dem maschinellen Lernen von Ontologien eine besondere Bedeutung zu. Wie vorher bereits beschrieben, kann durch den automatischen Erwerb von Ontologien die Möglichkeiten des Ontology Switching erweitert werden, so dass Benutzer verschiedene Perspektiven auf einen multimedialen Bestand auswählen können. Zur Erweiterung der Perspektiven in MyShelf würde sich daher eine semi-automatische Vorgehensweise anbieten. Zum einen kann viel manuelle Arbeit maschinengestützt durchgeführt werden und zum anderen wird das Domänenwissen des Experten korrekt eingesetzt, was eine zentrale Bedeutung für die korrekte Funktionalität einer MyShelf-Perspektive darstellt, da diese domänen-spezifisch fungieren soll. Nur so kann gewährleistet werden, dass das Domänenwissen innerhalb der Ontologie korrekt repräsentiert wird.

7 Zusammenfassung und Ausblick

In der vorliegenden Arbeit wurden verschiedene Ansätze zum maschinellen Lernen von Ontologien vorgestellt. Dabei wurden Versuche zum Lernen von Ontologien durchgeführt und die Ergebnisse evaluiert.

Aufgrund der durchgeführten Lernversuche stellte sich heraus, dass bei den vollautomatischen Ansätzen nur teilweise korrekte semantische Beziehungen von Konzepten in die Ontologie integriert wurden. Allerdings trat dies nur sehr vereinzelt auf, so dass eine vollautomatische Erstellung einer Ontologie keine benutzbaren bzw. benutzerfreundlichen Resultate bezüglich des Anwendungsfalles Browsing liefert. Diese Ergebnisse müssen von einem Experten auf jeden Fall geprüft und nachbereitet werden.

Ferner konnte gezeigt werden, dass bei semi-automatischen Ansätzen der Einsatz von Expertenwissen im Mittelpunkt steht. Allerdings stellen diese durchaus eine Unterstützung bei der Erstellung von Ontologien dar. Durch die Interaktion mit einem Experten kann Domänenwissen in die Ontologie eingebunden werden. Hierbei unterstützen maschinelle Verfahren den Experten bei der Auswahl von Konzepten, was den manuellen Aufwand für den Experten verringert. So können einzelne Schritte des Erstellungsprozesses einer Ontologie optimiert werden, wie zum Beispiel die Extraktion der Konzepte aus einem domänen-spezifischen Korpus oder die Generierung einer Auswahlliste mit Vorschläge von Konzeptkandidaten. Der Ansatz von OntoLT stellt diesbezüglich eine Möglichkeit dar, Ontologien semi-automatisch zu erstellen. Dadurch, dass die Mapping-Regeln um domänen-spezifische Konzepte erweitert werden, kann bei künftigen Extraktionen von Konzepten diese Erfahrungswerte weiterverwendet werden. Die Auswahl der Konzepte durch den Benutzer gewährleistet, dass semantische Zusammenhänge von Konzepten aus domänen-spezifischer Sicht in die Ontologie übertragen werden. Bei semi-automatischen Verfahren kommt Domänenwissen somit vom Menschen und gewährleistet korrekte semantische Zusammenhänge zwischen Konzepten. Dabei verspricht das halb-automatische Vorgehen bei der Erstellung von Ontologien nicht nur, den Aufwand für Experten zu verringern, sondern die Anzahl der Experten, die dazu benötigt werden, zu minimieren. Auch bietet es die Chance

hinsichtlich des zeitlichen Aufwands, dass es sich aus wirtschaftlicher Sicht für Wissensingenieure künftig lohnt, ihr Wissen Schritt für Schritt in eine Ontologie zu übertragen. Dies könnte wiederum Einfluss darauf haben, dass es später hinsichtlich des Einsatzes von Ontologien für Benutzer unterschiedliche Qualität von Informationen unterschiedliche Preise aufgrund der Erstellungskosten geben könnte.

Die Weiterentwicklung semi-automatischer Verfahren könnte die Schritte bei der Erstellung von Ontologien verbessern. Hierfür stehen durch die vorliegende Arbeit erschlossene Textkorpora zur Verfügung, die für weitere Versuche mit maschinellen Lernverfahren – seien es vollautomatische oder semi-automatische Vorgänge - verwendet werden können, um anhand anderer Ansätze zum Ontology Learning deren Potential zur Erstellung von Ontologien weiterhin testen zu können.

Auch Evaluierungsmethoden und –kriterien für maschinell erstellte Ontologien können erweitert und verfeinert werden, so dass die Qualität von Ontologien standardisiert untersucht werden kann. Allerdings sind hierbei insbesondere die Benutzerbedürfnisse und der gegebene Anwendungsbereich zu beachten.

Gerade für das Internet kommt Ontologien im Rahmen des Semantic Web eine zentrale Bedeutung zu. Wie FENSEL ET AL. (2003) beschreiben, bieten Ontologien verschiedene Einsatzmöglichkeiten im Internet, wie z.B. „intelligent search instead of keyword matching, query answering instead of information retrieval, document exchange among departments via ontology mappings, and definition of customized views on documents.“ So gesehen stellen Ontologien die nächste Stufe bei der Entwicklung des Semantic Web dar. Die Möglichkeit, dass Maschinen semantische Bezüge zwischen Informationen erkennen und auch verstehen können, bleibt vorerst Vision des Semantic Web.

Hinsichtlich MyShelf spielt das maschinelle Lernen von Ontologien eine besondere Bedeutung. Wie bereits beschrieben, kann durch den automatischen Erwerb von Ontologien die Möglichkeiten des Ontology Switching erweitert werden, so dass Benutzer verschiedene Perspektiven auf einen multimedialen Bestand auswählen können. Aufgrund der zunehmenden Digitalisierung von Informationen wird die Notwendigkeit des Einsatzes von maschinellem Lernen zunehmend an Bedeutung gewinnen. MyShelf bietet derzeit den Zugriff auf eine Momentaufnahme des Bücherbestandes der UB Hildesheim von 2002 und eine Auswahl an Websites für die Bereiche Information Retrieval und Computerlinguistik. Die

Darstellung der Bestände in MyShelf ist derzeit statisch. Mit Hilfe maschineller Lernverfahren jedoch kann eine darüber hinausgehende Flexibilität und Aktualität gewährleistet oder erreicht werden. Da immer mehr elektronische Dokumente neben der Recherche in einer Präsenzbibliothek hinzugezogen werden, vollzieht sich eine Entwicklung in Richtung Digital Library. Hierbei sind alle Dokumente in elektronischer Form zugänglich, so dass auf verteilte Dokumentenkollektionen ein virtueller Zugriff erfolgen kann. Der Einsatz von Ontologien könnte in diesem Zusammenhang gewährleisten, dass auf den gleichen Bestand jeweils eine andere spezifische Perspektive bei der Informationssuche ausgewählt werden kann, aber dennoch alle Medien berücksichtigt werden. Nur so kann sichergestellt werden, dass relevante Informationen, die ja nicht unbedingt aus einer Wissensdomäne stammen, einbezogen werden.

Die Hanke-Klassifikation erfüllt ihre Rolle als virtuelle Signatur für Informationswissenschaft an der UB Hildesheim. Eine Erweiterungsmöglichkeit dieser Klassifikation könnte so aussehen, dass elektronische Veröffentlichungen im Bereich Informationswissenschaft dem Bestand manuell hinzugefügt werden und die Ontologie daraufhin semi-automatisch angepasst wird, so dass eventuell neue Kategorien in die Hanke-Klassifikation aufgenommen werden. Dabei steht die Interaktion mit Experten wieder im Vordergrund, welche letztlich fachspezifische Entscheidungen treffen müssen. Auch ist hier die schrittweise Erweiterung der Mapping-Regel mit OntoLT eine Möglichkeit, die Hanke-Klassifikation zu erweitern, da Domänenwissen auf andere Korpora der gleichen Domäne übertragen wird. Eine weitere Möglichkeit wäre, dass ein Algorithmus die domänen-spezifischen Entscheidungen des Experten bei der semi-automatischen Vorgehensweise lernt bzw. mit diesen Entscheidungen trainiert wird. Somit könnte das Domänenwissen erlernt und auf andere Korpora der Domäne angewandt werden. Allerdings ist hierbei weiterhin eine anschließende manuelle Überprüfung der Ontologien durch den Experten erforderlich.

Insgesamt bieten semi-automatische Verfahren die Möglichkeit, dass sowohl bei der Verarbeitung der Informationen als auch bei dem Einsatz von Expertenwissen bei der Auswahl und Überprüfung der Ergebnisse jeweils eine optimale Ausrichtung erreicht wird. Maschinen sollen idealerweise so viele Informationen wie möglich derart auf- und vorbereiten können, so dass ein Experte sein Domänenwissen auf eine optimale Art und Weise einsetzen kann und der Arbeitsaufwand so minimal wie möglich gehalten wird. Insgesamt bleibt

festzustellen, dass die die maschinelle Leistung bei der Ontologieerstellung von der vergleichbaren menschlichen Performance noch weit entfernt ist:

One machine can do the work of fifty ordinary men.
No machine can do the work of one extraordinary man.

Elbert Green Hubbard in MOURSUND (2004)

Literaturverzeichnis

Alle hier und in Fußnoten aufgeführten Internetquellen (URLs) wurden am 15.03.2005 verifiziert.

[AGIRRE ET AL. 2001]

AGIRRE, ENEKO; ANSA, OLATZ; HOVY, EDUARD; MARTINEZ, DAVID (2001):
Enriching WordNet concepts with topic signatures. In: Proceedings NAACL WordNet Workshop, 2001.

[ANTONIOU; VAN HARMELEN 2003]

ANTONIOU, GRIGORIS; VAN HARMELEN, FRANK (2003):
Web Ontology Language: OWL. In: [STAAB; STUDER 2004], S.68-92.

[BERNERS-LEE 2000]

BERNERS-LEE, TIM (2000):
Semantic Web - XML2000 - Slide "Architecture". http://www.w3.org/2000/Talks/1206-xml2k-tbl/Overview.html

[BERNERS-LEE ET AL. 2001]

BERNERS-LEE, TIM; HENDLER, JAMES; LASSILA, ORA (2001):
The Semantic Web. In: Scientific American, May 2001, S. 29-37.

[BUITELAAR ET AL. 2003]

BUITELAAR, PAUL; OLEJNIK, DANIEL; SINTEK, MICHAEL (2003):
OntoLT: A Protégé Plug-In for Ontology Extraction from Text. In: Proceedings of ISWC2003: Demo Session, Sanibel Island, Florida, USA, October 21st, 2003.

[BUITELAAR ET AL. 2004]

BUITELAAR, PAUL; OLEJNIK, DANIEL; SINTEK, MICHAEL (2004):
A Protégé Plug-In for Ontology Extraction from Text Based on Linguistic Analysis. In: Proceedings of the 1st European Semantic Web Symposium (ESWS), Heraklion, Greece, May 2004, S. 31-44.

[CIMIANO ET AL. 2003]

CIMIANO, PHILIPP; STAAB, STEFFEN; TANE, JULIEN (2003):
Automatic Acquisition of Taxonomies from Text: FCA meets NLP. In: Proceedings of the ECML/PKDD Workshop on Adaptive Text Extraction and Mining, Cavtat-Dubrovnik, Croatia, S. 10-17.

[CIMIANO ET AL. 2004]

CIMIANO, PHILIPP; SCHMIDT-THIEME, LARS; PIVK, ALEKSANDER; STAAB, STEFFEN (2004):
Learning Taxonomic Relations from Heterogeneous Evidence. In: Proceedings of the ECAI 2004 Ontology Learning and Population Workshop.

[DECLERCK 2000]

DECLERCK, THIERRY (2002):
A set of tools for integrating linguistic and non-linguistic information. In: Proceedings of the SAAKM workshop at ECAI, Lyon, 2004.

[DOAN ET AL. 2004]

DOAN, ANHAI; MADHAVAN, JAYANT; DOMINGOS, PEDRO; HALEVY, ALON (2004):
Ontology Matching: A Machine Learning Approach. In: [STAAB; STUDER 2004], S.385-403.

[FAYYAD ET AL. 1996]

FAYYAD, USAMA; PIATETSKY-SHAPIRO, GREGORY; SMYTH, PADHRAIC (1996):
From Data Mining to Knowledge Discovery. In: AI Magazine 17(3): Fall 1996, S. 37-54.

[FENSEL ET AL. 2003]

FENSEL, DIETER; HENDLER, JAMES; LIEBERMANN, HENRY; WAHLSTER, WOLFGANG (2003):
Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential. Wahlster, Wolfgang (Hrsg.) MIT Press: Cambridge/London.

[FRAWLEY ET AL. 1992]

FRAWLEY, WILLIAM. J.; PIATETSKY-SHAPIRO, GREGORY; MATHEUS, CHRISTOPHER J. (1992):

Knowledge Discovery in Databases - An Overview. Ai Magazine, Volume 13, 1992, S.57-70.

[GANTER; WILLE 1999]

GANTER, BERNHARD; WILLE, RUDOLF (1999):

Formal Concept Analysis – Mathematical Foundations. Springer Verlag.

[GÓMEZ-PÉREZ 1999]

GÓMEZ-PÉREZ, ASUNCIÓN (1999):

Evaluation of Taxonomic Knowledge on Ontologies and Knowledge-Based Systems. In: Proceedings of the North American Workshop on Knowledge Acquisition, Modeling, and Management, KAW 1999.

[GOOGLE 2004]

GOOGLE (2004):

Google Web APIs – Home.

http://www.google.com/apis/

[GRUBER 1993]

GRUBER, THOMAS (1993):

A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition, Vol. 5, 1993, S. 199-220.

[HANKE 2002]

HANKE, PETER (2002):

Neue Chancen und Möglichkeiten für Ordnungssystematiken durch Virtualisierung: Anwendung am Beispiel der Erfassung und Klassifizierung des informationswissenschaftlichen Bücherbestandes der Universitätsbibliothek Hildesheim.

Magisterarbeit, Universität Hildesheim, Fachbereich III – Informations- und Kommunikationswissenschaften.

[HANKE ET AL. 2002]

HANKE, PETER; MANDL, THOMAS ; WOMSER-HACKER, CHRISTA (2002):
Ein "Virtuelles Bibliotheksregal" für die Informationswissenschaft als Anwendungsfall semantischer Heterogenität. In: Hammwöhner, Rainer; Wolff, Christian & Womser-Hacker, Christa (Hrsg.): Information und Mobilität: Optimierung und Vermeidung von Mobilität durch Information. Proceedings 8. Internationalen Symposiums für Informationswissenschaft. 7.-10. Oktober 2002, Regensburg. Konstanz: Universitätsverlag [Schriften zur Informationswissenschaft Bd. 40], S. 289-302.

[HARRIS 1968]

HARRIS, ZELLIG (1968):
Mathematical structures of language. John
Wiley & Sons, New York, US, 1968

[HEARST 1992]

HEARST, MARTI A. (1992):
Automatic acquisition of hyponyms from large text corpora. In Proceedings of the 14th International Conference on Computational Linguistics. Nantes, France, 1992, S. 539 - 545.

[HEINZ 2003]

HEINZ, SABINE (2003):
Realisierung und Evaluierung eines virtuellen Bibliotheksregals für die Informationswissenschaft an der Universitätsbibliothek Hildesheim. Magisterarbeit, Universität Hildesheim, Fachbereich III – Informations- und Kommunikationswissenschaften.

[HEINZ 2003a]

HEINZ, SABINE (2003):
MyShelf: Der informationswissenschaftliche Bestand der Universitätsbibliothek Hildesheim aus der Sicht dreier verschiedener Systematiken, sozusagen durch drei unterschiedliche "Brillen", betrachtet.

http://web1.bib.uni-hildesheim.de/edocs/2003/363197524/doc/VirtuellesBibliotheksregalderInformationswissenschaft/index.html

[HEINZ 2003b]

HEINZ, SABINE (2003):
MyShelf: Systematik B.
http://web1.bib.uni-hildesheim.de/edocs/2003/363197524/doc/VirtuellesBibliotheksregalderInformationswissenschaft/Systematik_B/index.html

[KÖLLE ET AL. 2004]

KÖLLE, RALPH; MANDL, THOMAS; SCHNEIDER, RÉNE; STRÖTGEN, ROBERT (2004):
Weiterentwicklung des virtuellen Bibliotheksregals MyShelf mit semantic web Technologie. Erste Erfahrungen mit informationswissenschaftlichen Inhalten. In: Ockenfeld, Marlies (Hrsg.): Information Professional 2011. Strategien - Allianzen - Netzwerke; 26. Online-Tagung der DGI, Frankfurt am Main, 15. bis 17. Juni 2004; Proceedings. Frankfurt am Main: DGI 2004. (Tagungen der Deutschen Gesellschaft für Informationswissenschaft und Informationspraxis; 6), S. 111-124.

[KORTMANN 1999]

KORTMANN, BERND (1999):
Linguistik: Essentials; Anglistik, Amerikanistik. Cornelsen: Berlin.

[LAVELLI ET AL. 2004]

LAVELLI, ALBERTO; SEBASTIANI, FABRIZIO; ZANOLI, ROBERTO (2004):
Distributional term representations: an experimental comparison. In: Conference on Information and Knowledge Management archive
Proceedings of the Thirteenth ACM conference on Information and knowledge management table of contents, Washington, D.C., USA, 2004, S. 615-624.

[LAWRENCE ET AL. 1999]

LAWRENCE, STEVE; GILES, C. LEE; BOLLACKER, KURT (1999):
Digital Libraries and Autonomous Citation Indexing.
IEEE Computer, Volume 32, Number 6, 1999, S. 67-71.

[LIEBSCH 2004]

LIEBSCH, FRANZISKA (2004):
RFID and the Semantic Web. In: XML Clearinghouse Report 10. Tolksdorf, R.(Hrsg.); Eckstein, R. (Hrsg.).

[LOPEZ 1999]

LOPEZ, FERNANDEZ (1999):
Overview of Methodologies for Building Ontologies. In: Proceedings of the IJCAI-99 workshop on Ontologies and Problem-Solving Methods, Stockholm, Sweden, August 2, 1999.

[MAEDCHE; STAAB 2001]

MAEDCHE, ALEXANDER; STAAB, STEFFEN (2001):
Ontology Learning for the Semantic Web. In: IEEE Intelligent Systems 16 (2). March 2001. Special Issue on Semantic Web. S. 72-79.

[MAEDCHE; STAAB 2002]

MAEDCHE, ALEXANDER.; STAAB, STEFFEN (2002):
Measuring Similarity between Ontologies. In: Proceedings of the European Conference on Knowledge Acquisition and Managment - EKAW-2002. Madrid, Spain, October 1-4 2002. LNCS/LNAIc 2473, Springer, S.251-263.

[MAEDCHE; STAAB 2004]

MAEDCHE, ALEXANDER; STAAB, STEFFEN (2004):
Ontology Learning. In: [STAAB; STUDER 2004], S. 173-189.

[MAEDCHE ET AL. 2001]

MAEDCHE, ALEXANDER; STAAB, STEFFEN; STUDER, RUDI (2001):
Ontologien. In: Wirtschaftsinformatik (2001): Band 43, Heft 4. Friedr. Vieweg & Sohn Verlagsgesellschaft mbH: Wiesbaden. S. 393-396.

[MAEDCHE ET AL. 2003]

MAEDCHE, ALEXANDER; PEKAR, VIKTOR; STAAB, STEFFEN (2003):
Ontology Learning Part One - On Discoverinh Taxonomic Relations from the Web. In: Ning Zongh et al. (Hrsgb.): Web Intelligence, Springer 2003, S. 301-320.

[MANDL; WOMSER-HACKER 2002]

MANDL, THOMAS; WOMSER-HACKER, CHRISTA (2002):
Die Virtuelle Signatur Informationswissenschaft als Modell für die benutzerorientierte Integration virtueller Dokumente in Bibliotheksbestände. In: The 8th Annual Meeting of the IuK Initiative Information and Communication of the Learned Societies in Germany "Offene Systeme für die Kommunikation in Wissenschaft und Forschung" Ulm, März, 10 - 13, 2002.

[MILLER 1995]

MILLER, GEORGE A. (1995):
WordNet: a lexical database for English. In: Communications of the ACM 38 (11), November 1995, S. 39-41.

[MITCHELL 1997]

MITCHELL, TOM M. (1997):
Machine Learning. New York et al.: McGraw-Hill.

[MORIK 1994]

MORIK, KATHARINA (1994):
Balanced Cooperative Modeling. In: Machine Learning - A Multistrategy Approach. Volume 11 ,Issue 2-3. Kluwer Academic Publishers: Hingham, MA, USA, S. 295-318.

[MORIK ET AL. 1993]

MORIK, KATHARINA; WROBEL, STEPHAN; KIETZ, JÖRG-UWE; EMDE, WERNER (1993):

Knowledge acquisition and machine learning: Theory, methods, and applications. Academic Press, London, 1993.

[MOURSUND 2004]

MOURSUND, DAVE (2004):

Planning, Forecasting, and Inventing Your Computers-in-Education Future.

http://darkwing.uoregon.edu/~moursund/InventingFutures/FuturesMSWord.doc

[NAISBITT 1982]

NAISBITT, JOHN (1982):

Megatrends: Ten New Directions Transforming Our Lives. Warner Books: New York.

[NEC RESEARCH INSTITUTE 2004]

NEC RESEARCH INSTITUTE (2004):

Computer and Information Science Papers CiteSeer Publications ResearchIndex.

http://citeseer.ist.psu.edu/.

[NEC RESEARCH INSTITUTE 2004a]

NEC RESEARCH INSTITUTE (2004):

About CiteSeer.

http://citeseer.ist.psu.edu/citeseer.html.

[NEC RESEARCH INSTITUTE 2004b]

NEC RESEARCH INSTITUTE (2004):

Computer Science Directory [CiteSeer; Steve Lawrence, Kurt Bollacker, Lee Giles; NEC Research Institute].

http://citeseer.ist.psu.edu/directory.html

[NLP GROUP 2005]

NATURAL LANGUAGE PROCESSING GROUP, UNIVERSITÄT SHEFFIELD (2005):
GATE - A General Architecture for Text Engineering.
http://gate.ac.uk/

[NOY, MCGUINESS 2001]

NOY, NATALYA F.;MCGUINNESS, DEBORAH L. (2001):
Ontology Development 101: A Guide to Creating Your First Ontology. In: Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, March 2001.

[OMELAYENKO 2001]

OMELAYENKO, BORYS (2001):
Learning of Ontologies for the Web: the Analysis of Existent Approaches. In: Proceedings of the International Workshop on Web Dynamics held in conj. with the 8th International Conference on Database Theory (ICDT'01), London, UK, January 3, 2001.

[SEIPEL; BAUMEISTER 2004]

SEIPEL, DIETMAR; BAUMEISTER, JOACHIM (2004):
Declarative Methods for the Evaluation of Ontologies. In: KI 4/2004, S. 51-57.

[STAAB; STUDER 2004]

STAAB, STEFFEN; STUDER, RUDI (2004):
Handbook on Ontologies - With 190 Figures and 22 Tables. Springer: Berlin et al.

[STAAB; STUDER 2004a]

STAAB, STEFFEN; STUDER, RUDI (2004):
Ontology Learning. In: [STAAB; STUDER 2004], S.173-189.

[STAAB ET AL. 2003]

STAAB, STEFFEN; STUMME, GERD; HARTMANN, JENS; TANE, JULIEN (2003): Folien zur Vorlesung Knowledge Discovery: Kapitel VII Unüberwachte Data Mining Verfahren - Assoziationsregeln für das Ontologielernen. http://www.aifb.uni-karlsruhe.de/Lehre/Winter2002-03/kdd/download/VII-4-Ontologielernen.pdf

[STAAB ET AL. 2003a]

STAAB, STEFFEN; STUMME, GERD; HARTMANN, JENS; TANE, JULIEN (2003): Folien zur Vorlesung Knowledge Discovery: Kapitel VII - Formale Begriffsanalyse. http://www.aifb.uni-karlsruhe.de/Lehre/Winter2002-03/kdd/download/VII.5_Formale_Begriffsanalyse.pdf

[TEXTTOONTO 2003]

TEXTTOONTO (2003): TextToOnto User Manual 2003. http://sf.gds.tuwien.ac.at/00-pdf/t/texttoonto/texttoonto.pdf

[VELARDI ET AL. 2001]

VELARDI, PAOLA; FABRIANI, PAOLO.; MISSIKOFF, MICHELE (2001): Using text processing techniques to automatically enrich a domain ontology. In: Proceedings of the ACM International Conference on Formal Ontology in Information Systems - Volume 2001, 2001. S. 270 - 284.

[W3C 2004]

WORLD WIDE WEB CONSORTIUM W3C (2004): OWL Web Ontology Language - Introduction. http://www.w3.org/TR/2004/REC-owl-guide-20040210/#Introduction

[W3C 2004a]

WORLD WIDE WEB CONSORTIUM W3C (2004A): OWL Web Ontology Language - Use Cases and Requirements. http://www.w3.org/TR/webont-req/

[WIEGEMANN 2004]

WIEGEMANN, SVENJA (2004):
Implementierung einer benutzungsfreundlichen Oberfläche für mobile Endgeräte am Beispiel eines Bibliotheksinformationssystems. Magisterarbeit, Universität Hildesheim, Fachbereich III – Informations- und Kommunikationswissenschaften.

[WILHELM 2004]

WILHELM, BETTINA (2004):
Der virtuelle Wegweiser Informationswissenschaft - Entwicklung und Implementierung eines Konzepts für die Integration eines Clearinghouse in das virtuelle Bibliotheksregal MyShelf. Magisterarbeit, Universität Hildesheim, Fachbereich III - Informations- und Kommunikationswissenschaften.

[WILHELM 2004a]

WILHELM, BETTINA (2004):
Virtuellen Wegweiser Informationswissenschaft - Browse.
http://www.vww-info.de/browse.php

[WITTEN ET AL. 2001]

WITTEN, IAN H.; FRANK, EIBE (2001):
Data Mining - Praktische Werkzeuge und Techniken für das maschinelle Lernen. Übers.: Märten, Dr. D; Muhr, J. .Carl Hanser Verlag: München, Wien.

Abbildungsverzeichnis

Tabellenverzeichnis

Anhang

Inhalt der DVDs

Auf den beiliegenden DVDs befinden sich folgende Inhalte:

DVD 1:

1. Dateien
- 1.1 Magisterarbeit (Pdf-Datei)
- 1.2 Lernergebnisse TextToOnto
- 1.3 Lernergebnisse OntoLT
- 1.4 OntoLT: Ergebnis der Konzepte nach Chi-Square-Analyse
- 1.5 Versuchsanordnungen der Lernergebnisse
- 1.6 Recall-Werte der Lernergebnisse (Excel-Datei)

2. Tools
- 2.1 MyCrawler v 1.0
- 2.2 KAON TextToOnto + WordNet 1.7.1
- 2.3 OntoLT + Protégé 2000 1.8
- 2.4 HTMLasText
- 2.5 Anawave WebSnake
- 2.6 PDF2Text
- 2.7 Jaws PDFCreator

3. Korpora
- 3.1 Übersicht CiteSeer (Excel-Datei)
- 3.2 CiteSeer: Computerlinguistik - erschlossene Dokumente mit PostScript etc.
- 3.3 CiteSeer: Information Retrieval repräsentative Auswahl konvertierter Dateien
- 3.4 CiteSeer: Alle konvertierten Dateien aus CL + IR
- 3.5 CiteSeer: Abstracts CL + IR
- 3.6 Übersicht Websites (Excel-Datei)
- 3.7 Erschlossene Englische Sites (Excel-Datei)
- 3.8 Datenbank-Websites: Alle konvertierten Dateien
- 3.9 Datenbank-Websites: Alle gesammelten Originaldateien
- 3.10 Datenbank: MySQL-Dump und Auflistungen

DVD 2:

1. Dateien
- 1.1 Übersicht CiteSeer-Kategorien

2. Korpora
- 2.1 CiteSeer: Information Retrieval - erschlossene Dokumente mit PostScript etc.

Anhang

Anhang I: Hanke-Klassifikation

Anhang II: Auswertung der Datenbank Links

Anhang III: CiteSeer Suchbegriffe und Verzeichnisse

Anhang IV: Ergebnis CiteSeer Downloads

Anhang V: Beispiel eines mit SCHUG annotierten Satzes

Anhang VI: TextToOnto Versuchsanordnungen

Anhang VII: Lernergebnis mit OntoLT

Anhang VIII: Recall-Berechnung der Ergebnisse von TextToOnto und OntoLT

Anhang IX: Handbuch zu MyCrawler

Anhang I: Hanke-Klassifikation nach HANKE (2002:Anlage 18) - Teil 1

1. Allgemein	2. Grundlagen / Theorie
1.1. Bibliographien 1.2. Schriften und Berichte 1.2.1. Schriftenreihen 1.2.2. Tagungs- und Kongreßberichte 1.2.3. Fest- und Gedenkschriften 1.2.4. Jahresberichte 1.2.5. Sonstiges 1.3. Lehrbücher 1.4. Sammelwerke 1.5. Nachschlagewerke, Darstellungen 1.5.1. Fachwörterbücher 1.5.2. Handbücher 1.5.3. Lexika 1.5.4. Einführungen, Abrisse 1.5.5. Gesamtdarstellungen 1.5.6. Darstellungen zu mehreren Gebieten 1.5.7. Adressbücher 1.5.8. Sprachwörterbücher 1.5.9. Bilder, Tafelwerke 1.5.10. Tabellen, Formelsammlungen 1.5.11. Abkürzungen, Symbole 1.5.12. Bezugsquellenverzeichnisse, Preislisten 1.5.13. Sonstiges 1.6. Zeitschriften 1.7. Forschungseinrichtungen und -projekte 1.8. Institutionen, Organisationen und Verbände 1.9. Beruf, Studium und Ausbildung 1.9.1. Berufs- und Personalfragen 1.9.2. Ausbildungsstätten 1.9.3. Berufsbilder 1.9.4. Sonstiges 1.10. Geschichte 1.11. Biographien 1.12. Firmen	2.1. Informationstheorie 2.2. Kommunikationstheorie und -modelle 2.3. Fuzzy-Theorie 2.4. Informationspsychologie 2.4.1. menschliche Informationsverarbeitung 2.4.2. kognitive Modelle 2.4.3. Sonstiges 2.5. Semantik. Semiotik 2.6. Terminologie 2.6.1. Fachsprachen 2.6.2. Sonstiges 2.7. Thesauruskunde 2.8. Deskription 2.9. Inhaltliche Erschließung 2.9.1. Analysieren 2.9.2. Klassifizieren 2.9.3. Indexieren 2.9.4. Referieren 2.9.5. Sonstiges 2.10. Klassifikation 2.11. Dokumentationstypen 2.12. Normen (UML, SGML, XML u.a.) 2.13. Sonstiges

Anhang I: Hanke-Klassifikation - Teil 2

3. Informationstechnik / Elektronische Datenverarbeitung

3.1. Hardware
- 3.1.1. Datenspeicher, Datenträger
- 3.1.2. Rechner
  - 3.1.2.1. Personal Computer
  - 3.1.2.2. Datenbankrechner
  - 3.1.2.3. Großrechner
  - 3.1.2.4. Sonstiges
- 3.1.3. Periphere Geräte
  - 3.1.3.1. Eingabegeräte
  - 3.1.3.2. Ausgabegeräte
  - 3.1.3.3. Dialoggeräte
  - 3.1.3.4. Sonstiges
- 3.1.4. Sonstiges

3.2. Betriebssysteme

3.3. Dienstprogramme
- 3.3.1. Kalkulationsprogramme, Tabellenkalkulation
- 3.3.2. Textverarbeitungsprogramme
- 3.3.3. Software für Projekt-Management
- 3.3.4. Integrierte Software-Pakete
- 3.3.5. Desktop-Management-Systeme
- 3.3.6. Compiler
- 3.3.7. Spezielle Dienstprogramme
- 3.3.8. Sonstiges

3.4. Programmierung
- 3.4.1. Software Engineering
- 3.4.2. Programmiersprachen
  - 3.4.2.1. Assembler / Makrosprachen
  - 3.4.2.2. Objekt-orientierte Programmiersprachen
  - 3.4.2.3. Prozedurale Programmiersprachen
  - 3.4.2.4. Logische Programmiersprachen
  - 3.4.2.5. Entwicklungsumgebungen (Authorware, Powerbuilder, Visual Age u.a.)
  - 3.4.2.6. Sonstiges
- 3.4.3. Sonstiges und Theorie

3.5. Formale Sprachen (ohne Programmiersprachen)

3.6. Datenerfassung und -speicherung

3.7. Datenverwaltung, EDV-Management

3.8. Datenbanken
- 3.8.1. Datenmodellierung
- 3.8.2. Datenbanksprachen
- 3.8.3. Datenbanksysteme
- 3.8.4. Relationale Datenbanken
- 3.8.5. Objektorientierte Datenbanken
- 3.8.6. Multimediale Datenbanken
- 3.8.7. Textdatenbanken
- 3.8.8. Mediendatenbanken
- 3.8.9. Faktendatenbanken
- 3.8.10. Sonstiges

3.9. Datensicherung
- 3.9.1. Kryptographie
- 3.9.2. Computer-Viren
- 3.9.3. Sonstiges

3.10. Datenkomprimierung

3.11. Datennetze, Internet
- 3.11.1. Intranet
- 3.11.2. Netzwerke
- 3.11.3. Breitbandnetze, ISDN
- 3.11.4. Mobiles Computing, Mobilfunknetze
- 3.11.5. Protokolle
- 3.11.6. Internet-Server
- 3.11.7. Browser
- 3.11.8. Datenfernübertragung
- 3.11.9. Sonstiges

3.12. Rechenzentren

3.13. Sonstiges

Anhang I: Hanke-Klassifikation - Teil 3

4. Information Retrieval	5. Informationsqualität und Evaluierung
4.1. Retrievalsysteme und Dialogsprachen 4.2. Modelle des Information Retrieval 4.3. Suchmaschinen und Suchmethoden im Internet 4.4. Suchverfahren 4.5. Software-Agenten 4.6. Data-Mining, Knowledge Discovery 4.7. Sonstiges	5.1. Evaluierung von IR-Systemen 5.2. Qualitätsmanagement und Evaluierung informationeller Systeme 5.3. Qualität im Internet 5.4. Sonstiges

6. Informationsmanagement / Wissensmanagement	7. Informationswirtschaft / Informationsmarkt
6.1. Betriebliche Information und Kommunikation 6.2. Planung, Organisation 6.3. Dienstleistungen, Outsourcing 6.4. Innovationsforschung 6.5. Implementierungsforschung 6.6. Marketing 6.7. Büroautomatisierung 6.8. Kosten, Nutzen, Leistung, Gebühren, Controlling 6.9. Public-Relations-Arbeit 6.10. Projektmanagement 6.11. Entscheidungssysteme 6.12. Internationales Informationsmanagement 6.13. Sonstiges	7.1. Elektronische Märkte 7.2. Electronic Banking 7.3. Informationskosten 7.4. Informationsmarketing 7.5. Information brokering 7.6. Wirtschaftsinformationsquellen 7.7. Sonstiges

8. Human Computer Interaction	9. Sprachtechnologie
8.1. Software-Ergonomie 8.2. Dialogsysteme 8.3. Web-Design 8.4. Usability 8.5. Hypermedia / Multimedia 8.6. Sonstiges	9.1. Maschinelle und maschinengestützte Übersetzung 9.2. Computerlinguistik 9.3. Parsing 9.4. Maschinelle Sprachverarbeitung / Spracherkennung 9.5. Formale Grundlagen 9.6. Sonstiges

Anhang I: Hanke-Klassifikation - Teil 4

10. Computer Mediated Communication	11. Informationsvisualisierung
10.1. Electronic Mailing 10.2. Kommunikationssoftware 10.3. Kommunikationsforen 10.4. Sonstiges	11.1. Computergraphik, Grafik-Formate 11.2. Graphik-Programmpakete 11.3. Bildbearbeitung 11.4. Digitale Fotografie, Digitaler Film 11.5. Computer-Animation, virtuelle Welten 11.6. Computer-Simulation 11.7. Computer aided design 11.8. Sonstiges

12. Wissensvermittlung / Informations- und Dokumentationsstellen	13. Informations- und Dokumentationssysteme
12.1. Computerbasiertes Lernen und Unterrichten 12.2. Digitale Bibliotheken 12.3. Elektronisches Publizieren 12.3.1. Web Publishing 12.3.2. Desktop Publishing 12.3.3. Sonstiges 12.4. Elektronische Enzyklopädien 12.5. Verlage 12.5.1. Publishing on demand 12.5.2. E-Books, E-Journals 12.5.3. Sonstiges 12.6. Medien 12.6.1. Massenmedien 12.6.2. Elektronische Medien 12.6.3. Sonstiges 12.7. Bibliothekswesen, Dokumentationswesen 12.7.1. Bibliotheksautomatisierung 12.7.2. Katalogisierung, Erschließung 12.7.3. Sonstiges 12.8. Dokumentenmanagementsysteme, Archivsysteme 12.9. Sonstiges	13.1. Analytische Informationssysteme 13.2. Informationssysteme verschiedener Sachgebiete 13.2.1. Mathematik und Naturwissenschaften 13.2.2. Sozialwissenschaften 13.2.3. Geisteswissenschaften 13.2.4. Sonstiges 13.3. Sonstiges

Anhang I: Hanke-Klassifikation - Teil 5

14. Künstliche Intelligenz	15. Information und Gesellschaft / Informationspolitik
14.1. Mustererkennung, Zeichenerkennung 14.2. Lernende Systeme und Anpassungssysteme (incl. Neuronale Netze) 14.3. Automatisches Schließen, Reasoning 14.4. Simulation von Denkvorgängen 14.5. Diagnose 14.6. Expertensysteme 14.7. Sonstiges	15.1. Informationsrecht 15.1.1. Urheberrecht (Copyright) 15.1.2. Datenschutz 15.1.3. Sonstiges 15.2. Informationsethik 15.3. Informationsgesellschaft 15.4. Informationsverhalten und Benutzerforschung 15.5. Informationsbedürfnisse 15.6. Wirkungsforschung 15.7. Sozialwissenschaftliche Methoden 15.8. Förderungsprogramme 15.9. Sonstiges

16. Informationswissenschaftliche Anwendung in anderen Sachgebieten
16.1. Mathematik und Naturwissenschaften 16.2. Sozialwissenschaften 16.3. Geisteswissenschaften 16.4. Sonstiges

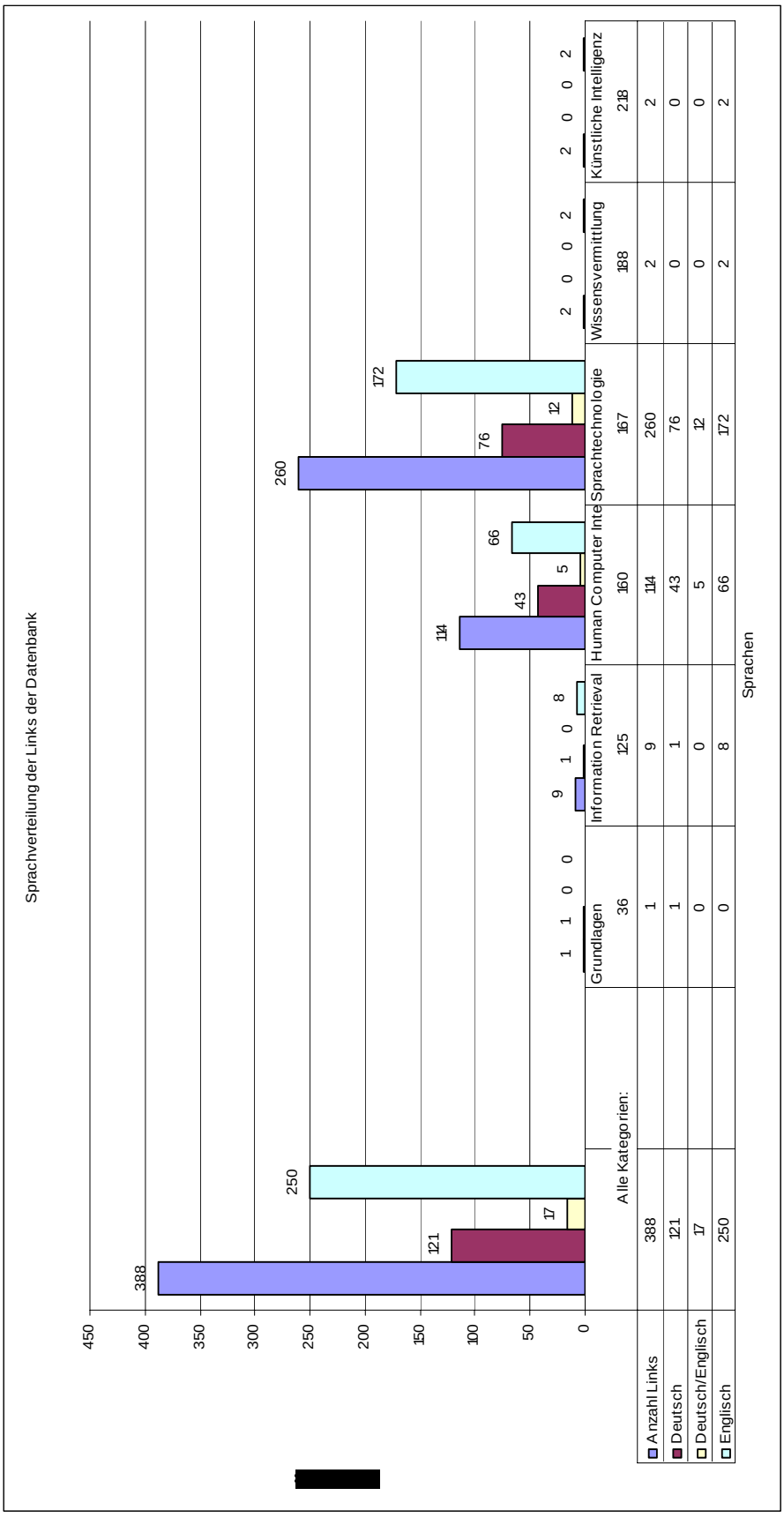
Anhang II: Auswertung der Datenbank Links - Teil 1

Kategorie	Kategoriename	Anzahl Links	Deutsch	Deutsch/Englisch	Englisch
Gesamt:		**347**	**107**	**17**	**223**
Grundlagen/ Theorie (36)		**1**	0	0	0
53	Indexieren	**1**	0	0	0
Information Retrieval (125)		**7**	0	0	0
126	Retrievalsysteme und Dialogsprachen	**6**	0	0	0
127	Modelle des Information Retrieval	**1**	0	0	0
Informationswirtschaft/ Informationsmarkt (152)					
153	Elektronische Märkte	**0**			
154	Electronic Banking	**0**			
155	Informationskosten	**0**			
156	Informationsmarketing	**0**			
157	Information brokering	**0**			
158	Wirtschaftsinformationsquellen	**0**			
159	Sonstiges	**0**			
Human Computer Interaction (160)		**114**	0	0	0
161	Software-Ergonomie	**23**	0	0	0
162	Dialogsysteme	**33**	0	0	0
163	Web-Design	**22**	0	0	0
164	Usability	**24**	0	0	0
165	Hypermedia / Multimedia	**11**	0	0	0
166	Sonstiges	**1**	0	0	0

Anhang II: Auswertung der Datenbank Links - Teil 2

Kategorie	Kategoriename	Anzahl Links	Deutsch	Deutsch/Englisch	Englisch
Sprachtechnologie (167)		**196**	0	1	0
168	Maschinelle und maschinengestützte Übersetzung	**11**	0	0	0
169	Computerlinguistik	**76**	0	1	0
170	Parsing	**30**	0	0	0
171	Maschinelle Sprachverarbeitung / Spracherkennung	**67**	0	0	0
172	Formale Grundlagen	**12**	0	0	0
173	Sonstiges	**0**			
Wissensvermittlung/ Informations-und Dokumentationsstellen (188)		**2**	0	0	0
189	Computerbasiertes Lernen und Unterrichten	**1**	0	0	0
190	Digitale Bibliotheken	**1**	0	0	0
Künstliche Intelligenz (218)		**2**	0	0	0
218	Künstliche Intelligenz	**2**	0	0	0

Anhang II: Auswertung der Datenbank Links - Teil 3

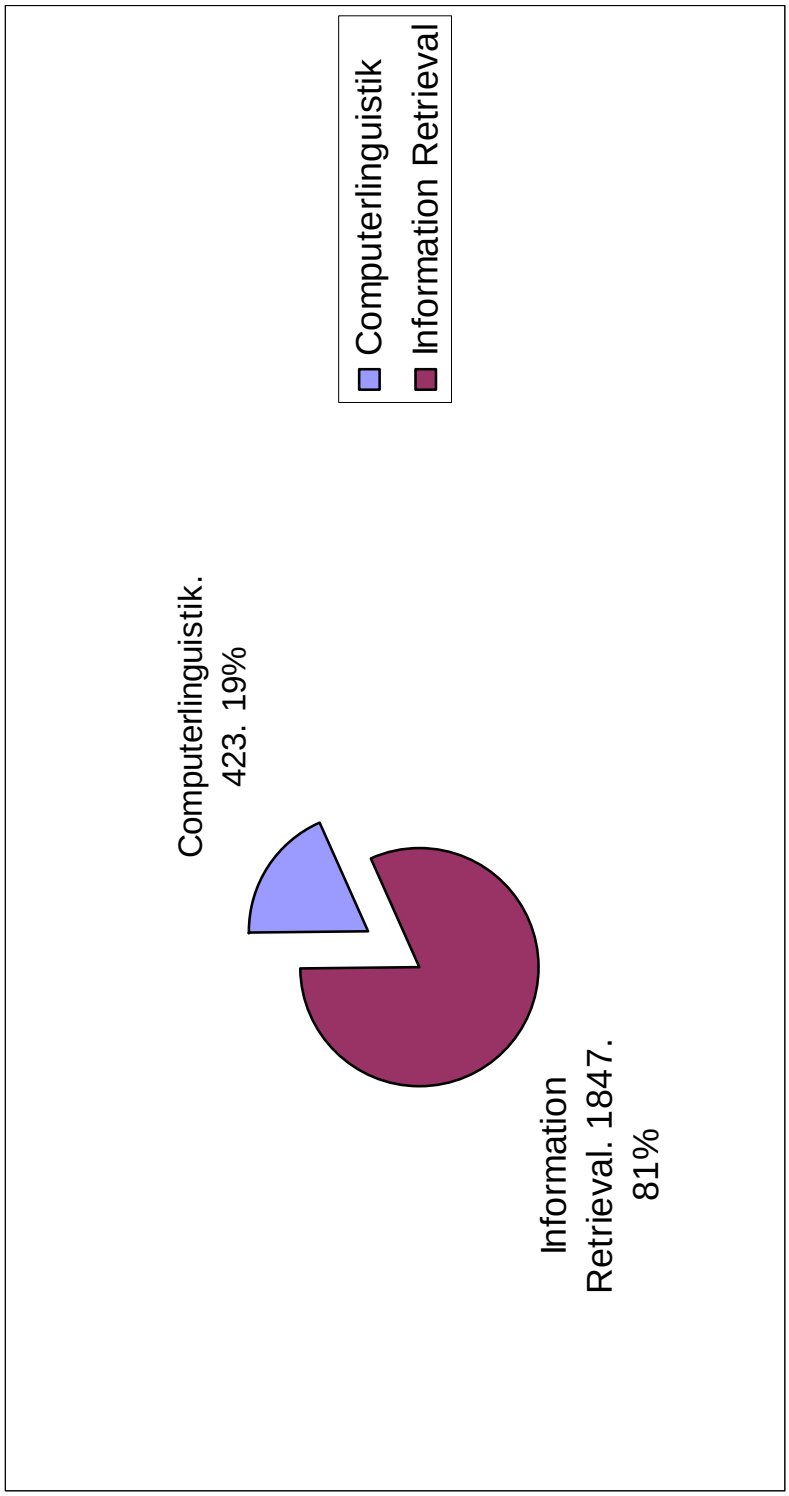


Anhang III: CiteSeer: Suchbegriffe und Verzeichnisse - Teil 1

Bereich	Art	Ordnername
Computerlinguistik	Suchbegriffe	computer+aided+translation
Computerlinguistik	Suchbegriffe	computer+linguistics
Computerlinguistik	Suchbegriffe	machine+translation
Computerlinguistik	Verzeichnis	NaturalLanguageProcessing

Bereich	Art	Ordnername
Information Retrieval	Suchbegriffe	data+mining
Information Retrieval	Suchbegriffe	information+extraction
Information Retrieval	Suchbegriffe	information+filtering
Information Retrieval	Suchbegriffe	information+retrieval
Information Retrieval	Verzeichnis	Classification
Information Retrieval	Verzeichnis	DigitalLibraries
Information Retrieval	Verzeichnis	Extraction
Information Retrieval	Verzeichnis	Filtering
Information Retrieval	Verzeichnis	InformationRetrieval
Information Retrieval	Verzeichnis	Metasearch
Information Retrieval	Verzeichnis	Retrieval
Information Retrieval	Verzeichnis	SearchEngines
Information Retrieval	Verzeichnis	WorldWideWeb

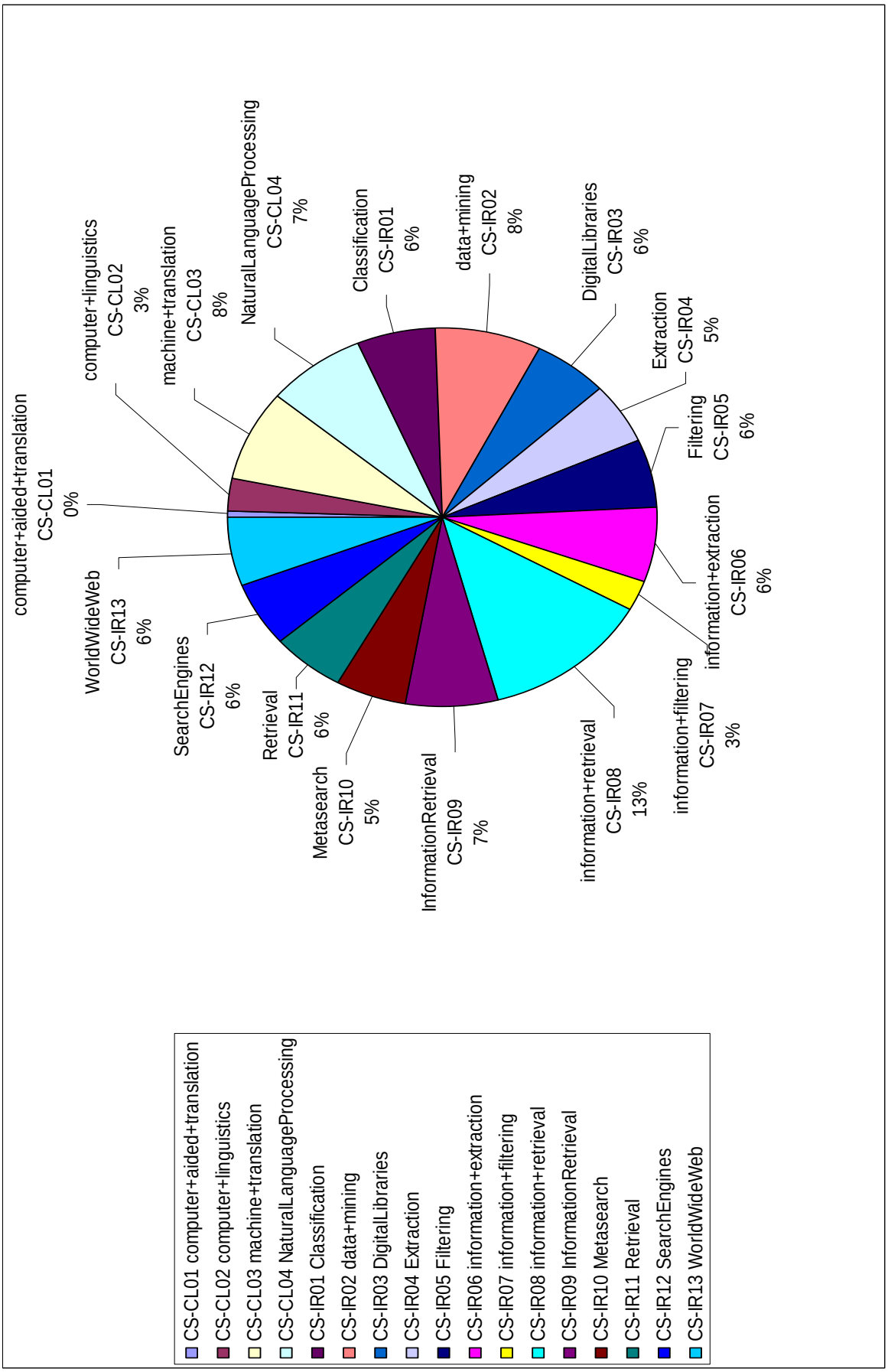
Anhang III: CiteSeer: Suchbegriffe und Verzeichnisse - Teil 2



Anhang IV: Ergebnis CiteSeer-Downloads - Teil 1

ID	Kategorie	Bereich	Ordnername	Anzahl Textdokumente
CS-CL01	CiteSeer	Computerlinguistik	computer+aided+translation	11
CS-CL02	CiteSeer	Computerlinguistik	computer+linguistics	66
CS-CL03	CiteSeer	Computerlinguistik	machine+translation	178
CS-CL04	CiteSeer	Computerlinguistik	NaturalLanguageProcessing	168
			Gesamt:	**423**
CS-IR01	CiteSeer	Information Retrieval	Classification	129
CS-IR02	CiteSeer	Information Retrieval	data+mining	185
CS-IR03	CiteSeer	Information Retrieval	DigitalLibraries	129
CS-IR04	CiteSeer	Information Retrieval	Extraction	115
CS-IR05	CiteSeer	Information Retrieval	Filtering	135
CS-IR06	CiteSeer	Information Retrieval	information+extraction	145
CS-IR07	CiteSeer	Information Retrieval	information+filtering	58
CS-IR08	CiteSeer	Information Retrieval	information+retrieval	290
CS-IR09	CiteSeer	Information Retrieval	InformationRetrieval	158
CS-IR10	CiteSeer	Information Retrieval	Metasearch	117
CS-IR11	CiteSeer	Information Retrieval	Retrieval	127
CS-IR12	CiteSeer	Information Retrieval	SearchEngines	127
CS-IR13	CiteSeer	Information Retrieval	WorldWideWeb	132
			Gesamt:	**1847**
			Insgesamt:	**2270**

Anhang IV: Ergebnis CiteSeer-Downloads - Teil 2



Anhang V: Beispiel eines mit SCHUG annotierten Satzes

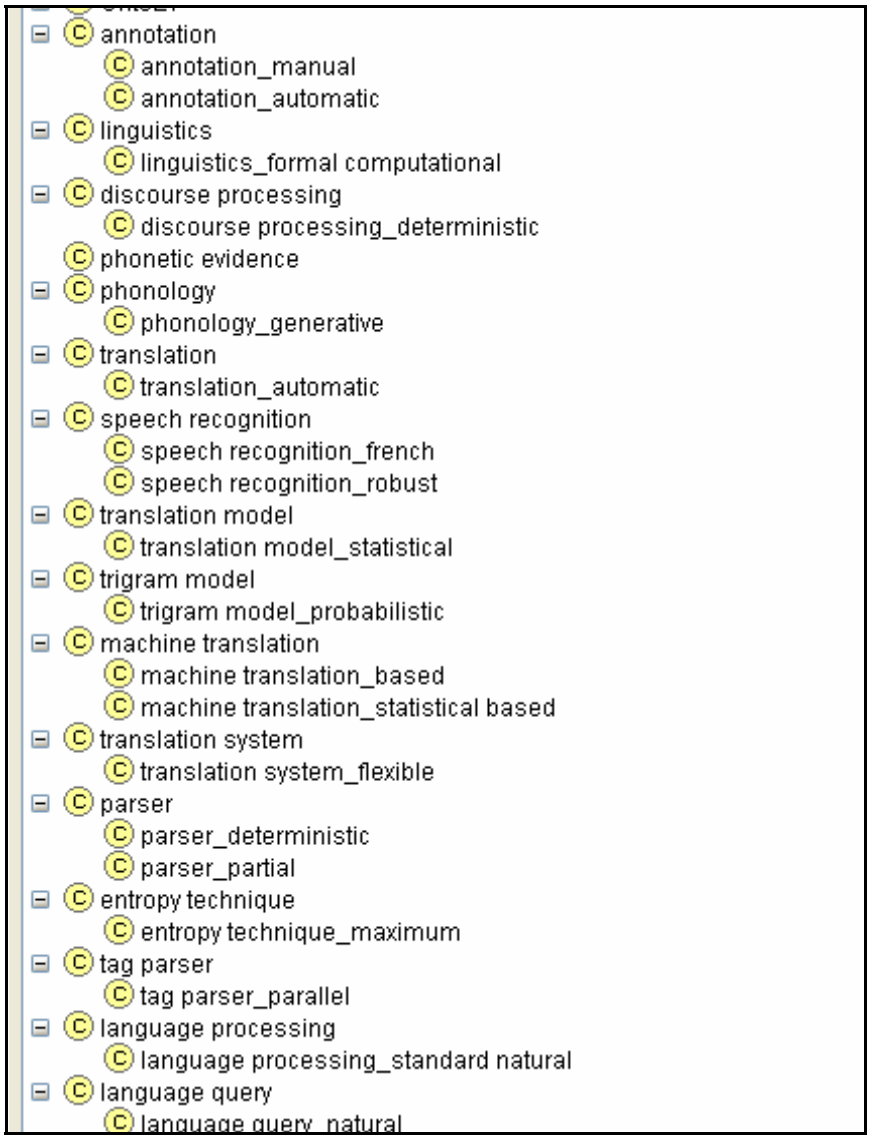
„An 40 Kniegelenkpräparaten wurden mittlere Patellarsehnendrittel mit einer neuen Knochenverblockungstechnik in einem zweistufigen Bohrkanal bzw. mit konventioneller Interferenzschraubentechnik femoral fixiert.“ [vgl. BUITELAAR ET AL. 2004:5]

```
<sentence id="s3" stype="decl" corresp=" ">

 <clauses>
  <clause id="cl1" from="p1" to="p5" pred="p5" type="pass">
   <arg id="a1" type="SUBJ" phrase="none" />
   <arg id="a2" type="IOBJ" phrase="p1"/>
   <arg id="a3" type="DOBJ" phrase="p2" />
   <arg id="a4" type="PP_ADJ" phrase="p3"/>
  </clause>
 </clauses>

 <phrases>
   ...
  <phrase id="p2" from="t5" to="t10" type="NP">
   <mod from="t5" to="t5" />
   <head from="t6" to="t6" />
   <mod_post from="t7" to="t10" />
  </phrase>
   ...
 </phrases>

 <text>
  <token id="t1" pos="APPR" str="An">
   <lemma id="t1.l1">an</lemma>
  </token>
  <token id="t2" pos="CARD" str="40" />
  <token id="t3" pos="NN" str="Kniegelenkpraeparaten">
   <lemma id="t3.l1">Kniegelenk</lemma>
   <lemma id="t3.l2">Praeparat</lemma>
  </token>
  <token id="t4" pos="VAFIN" str="wurden">
   <lemma id="t4.l1">werden</lemma>
  </token>
  <token id="t5" pos="ADJA" str="mittlere">
   <lemma id="t5.l1">mittler</lemma>
  </token>
  <token id="t6" pos="NN" str="Patellarsehnendrittel">
   <lemma id="t6.l1">patellar</lemma>
   <lemma id="t6.l2">Sehne</lemma>
   <lemma id="t6.l3">Drittel</lemma>
  </token>
   ...
  <token id="t19" pos="ADJD" str="femoral" />
  <token id="t20" pos="VVPP" str="fixiert">
   <lemma id="t6.l1">fixieren</lemma>
  </token>
  <token id="t21" pos="PUNCT" str="." />
 </text>
</sentence>
```

Anhang VI: TextToOnto Versuchsanordnungen

Nr.	Bestandteil	Anzahl	Parameter	Kommentar	Dateiname	Ergebnis
001	CiteSeer - alle	2270	Hearst, Heuristics, WordNet, 500	alle CiteSeer	modell_01.kaon	Absturz
002	CiteSeer - Computerlinguistik	423	Hearst, Heuristics, WordNet, 500	CiteSeer Computerlinguistik	modell_02.kaon	ok
003	CiteSeer - Computerlinguistik	423	Hearst, Heuristics, WordNet, 150	CiteSeer Computerlinguistik	modell_03.kaon	ok
004	CiteSeer - Information Retrieval	1847	Hearst, Heuristics, WordNet, 500	alle CiteSeer IR	modell_04.kaon	Absturz
005	CiteSeer - Information Retrieval, 1-7	896	Hearst, Heuristics, WordNet, 250	CiteSeer IR1-7	modell_05.kaon	Absturz
006	CiteSeer - Information Retrieval, 8-13	951	Hearst, Heuristics, WordNet, 500	CiteSeerIR 8-14	modell_06.kaon	Absturz
007	CiteSeer - Computerlinguistik - Abstracts	396	Hearst, Heuristics, WordNet, 500	CiteSeer - Computerlinguistik: Abstracts	modell_07.kaon	ok
007a	CiteSeer - Computerlinguistik - Abstracts	396	Hearst, Heuristics, WordNet, 50	CiteSeer - Computerlinguistik: Abstracts	modell_07a.kaon	ok
007b	CiteSeer - Computerlinguistik - Abstracts	396	Hearst, Heuristics, WordNet, 150	CiteSeer - Computerlinguistik: Abstracts	modell_07b.kaon	ok
008	CiteSeer - Computerlinguistik - Abstracts	396	FCA, 50, All Verbs	CiteSeer - Computerlinguistik: Abstracts	modell_08.kaon	ok
009	CiteSeer - Computerlinguistik - Abstracts	396	FCA, 50, Lexicographer Classe	CiteSeer - Computerlinguistik: Abstracts	modell_09.kaon	ok
010	CiteSeer - Datenbank	676	Hearst, Heuristics, WordNet, 500	CiteSeer - Datenbank: 2-30	modell_10.kaon	Absturz
011	CiteSeer - Information Retrieval 400	427	Hearst, Heuristics, WordNet, 50	CiteSeer - Information Retrieval 427	modell_11.kaon	ok
012	CiteSeer - Information Retrieval 400	427	Hearst, Heuristics, WordNet, 150	CiteSeer - Information Retrieval 427	modell_12.kaon	ok
013	CiteSeer - Information Retrieval 400	427	Hearst, Heuristics, WordNet, 500	CiteSeer - Information Retrieval 427	modell_13.kaon	ok
014	CiteSeer - Information Retrieval Abstracts	426	Hearst, Heuristics, WordNet, 150	CiteSeer - Information Retrieval Abstracts	modell_14.kaon	ok
015	CiteSeer - Information Retrieval Abstracts	426	Hearst, Heuristics, WordNet, 500	CiteSeer - Information Retrieval Abstracts	modell_15.kaon	ok
016	CiteSeer - Information Retrieval Abstracts	426	FCA, 50, All Verbs	CiteSeer - Information Retrieval Abstracts	modell_16.kaon	ok
017	CiteSeer - Information Retrieval Abstracts	426	FCA, 50, Lexicographer Classe	CiteSeer - Information Retrieval Abstracts	modell_17.kaon	ok
018	CiteSeer - CL + IR Abstracts	822	Hearst, Heuristics, WordNet, 50	CiteSeer - CL + IR Abstracts	modell_18.kaon	ok
019	CiteSeer - CL + IR Abstracts	822	Hearst, Heuristics, WordNet, 150	CiteSeer - CL + IR Abstracts	modell_19.kaon	ok
020	CiteSeer - CL + IR Abstracts	822	Hearst, Heuristics, WordNet, 500	CiteSeer - CL + IR Abstracts	modell_20.kaon	ok
021	CiteSeer - CL + IR Abstracts	822	FCA, 50, All Verbs	CiteSeer - CL + IR Abstracts	modell_21.kaon	ok
022	CiteSeer - CL + IR Abstracts	822	FCA, 50, Lexicographer Classe	CiteSeer - CL + IR Abstracts	modell_22.kaon	ok
023	CiteSeer - CL + IR Abstracts	822	Hearst, Heuristics, 500	ohne WordNet	modell_23.kaon	ok
024	CiteSeer - CL + IR Abstracts	822	Heuristics, WordNet, 50	ohne Hearst	modell_24.kaon	ok
025	CiteSeer - CL + IR Abstracts	822	Hearst, WordNet, 50	ohen Heuristiken	modell_25.kaon	ok

Anhang VII: Lernergebnis mit OntoLT



Vgl. DVD 1 – Ablage 1.3 *Lernergebnis-OntoLT.pprj*

Anhang VIII: Recall-Berechnung der Ergebnisse von TextToOnto und OntoLT- Teil 1

	modell_02	modell_03	modell_07	modell_07a	modell_07b	modell_08	modell_11	modell_12	modell_13	modell_14	modell_15	modell_16
CL (13 Stück)	**8**	**1**	**11**	**2**	**7**	**2**						
annotation	1		1		1							
artificial intelligence/ai	1		1									
computational linguistics												
language technology												
linguistics	1		1		1							
machine translation	1		1	1	1	1						
natural language processing / nlp	1		1		1							
parsing / parser	1		1		1							
semantics	1	1	1	1	1	1						
speech recognition			1									
syntax	1		1		1							
tagger			1									
word sense disambiguation			1									
IR (15 Stück)							**2**	**3**	**4**	**6**	**11**	**2**
classification										1	1	
data mining												
digital libraries												
extraction										1	1	
filtering												
information retrieval							1	1	1	1	1	
knowledge discovery											1	
meta search											1	
retrieval system												
search engine										1	1	1
recall											1	
precision								1	1	1	1	
software agents / agents									1	1	1	1
information system											1	
relevance feedback							1	1	1		1	

alle relevanten Konzepte:	13	13	13	13	13	13	15	15	15	15	15	15
enthaltene relevante Konzepte in der Ontologie:	8	1	11	2	7	2	2	3	4	6	11	2
alle relevanten Konzepte, die nicht in der Ontologie enthalten sind:	5	12	2	11	6	11	13	12	11	9	4	13
Recall:	**0,615**	**0,077**	**0,846**	**0,154**	**0,538**	**0,154**	**0,133**	**0,200**	**0,267**	**0,400**	**0,733**	**0,133**

Anhang VIII: Recall-Berechnung der Ergebnisse von TextToOnto und OntoLT - Teil 2

	modell_18	modell_19	modell_20	modell_21	modell23	modell_24	modell_25	modell_26		OntoLT
CL (13 Stück)	**0**	**2**	**10**	**0**	**3**	**0**	**0**	**0**		**8**
annotation			1							1
artificial intelligence/ai			1							
computational linguistics										
language technology										
linguistics			1							1
machine translation		1	1		1					1
natural language processing / nlp			1		1					1
parsing / parser			1							1
semantics		1	1							
speech recognition			1		1					1
syntax			1							1
tagger										1
word sense disambiguation			1							
IR (15 Stück)	**1**	**3**	**8**	**0**	**5**	**1**	**1**	**0**		
classification			1		1					
data mining										
digital libraries										
extraction			1		1					
filtering										
information retrieval		1	1		1					
knowledge discovery			1		1					
meta search										
retrieval system										
search engine		1	1		1					
recall										
precision			1							
software agents / agents	1	1	1			1	1			
information system										
relevance feedback			1							

alle relevanten Konzepte:	28	28	28	28	28	28	28	28		13
enthaltene relevante Konzepte in der Ontologie:	1	5	18	0	8	1	1	0		8
alle relevanten Konzepte, die nicht in der Ontologie enthalten sind:	27	23	10	28	20	27	27	28		5
Recall:	**0,036**	**0,179**	**0,643**	**0,000**	**0,286**	**0,036**	**0,036**	**0,000**		**0,615**

Anhang IX - Handbuch zu MyCrawler

MyCrawler ermöglicht es, Dokumente von CiteSeer automatisch zu erschließen. Dabei wird, wie auf der Internetseite von CiteSeer.com, ein Suchbegriff eingegeben und die Treffer der Suchanfrage nacheinander heruntergeladen. Da die URLs für die PostScript-Dateien auf der zweiten Ebene liegen, also nicht direkt die Suchtreffer darstellen, bedarf es einer weiteren Vorgehensweise. Nachdem Suchtreffer gefunden wurden, wird jeder einzelne Treffer weiterverfolgt und die PostScript-Datei jedes Artikels heruntergeladen.

Es werden folgende Funktionen von MyCrawler beschrieben:

1. Grundlegender Aufbau
2. Erschließung von CiteSeer-Dokumenten
3. Konvertierung von PDF-Dokumenten

1. Grundlegender Aufbau

1.1 Systemvoraussetzungen

Zur Programmierung von MyCrawler wurden *Eclipse Version: 3.0.0* und das *JDK in der Version 1.4.2_05* verwendet. Die Javadokumentation befindet sich im Ordner *MyCrawler Version 1.0\doc*. Alle verwendeten Klassenbibliotheken befinden sich im Ordner *MyCrawler Version 1.0\jar*.

1.2 Starten der Anwendung

In dem Ordner *MyCrawler Version 1.0* befindet sich die Batch-Datei *run.bat* zum Starten der Applikation, in welcher auch alle Klassenpfade gesetzt werden. Da für die Benutzeroberfläche

ein Windows LookandFeel eingesetzt wurde, kann es vorkommen, dass MyCrawler auf anderen Betriebssystemen als Windows Probleme macht. Hierzu kann man in der Datei Start.java, welche auch die Main-Methode enthält, den angegebenen Code für das LookAndFeel auskommentieren und die Applikation ohne jenes verwenden.

1.3 Die Benutzeroberfläche

Nach dem Starten von MyCrawler erscheint die Benutzeroberfläche im Tab Start. Die Oberfläche ist wie folgt aufgebaut:

- **Menüleiste**
  - **Datei**
    - *Neustart*: Mit dieser Funktion kann das Programm neu gestartet werden.
    - *Beenden:* Beendet die Applikation. Alternativ auch durch die Tastenkombination STRG+X.
  - **Optionen**
    - *Excel-Tab anzeigen*: Zeigt das Excel-Tab an, was standardmäßig inaktiv ist, da es nicht weiter implementiert wurde.
    - *Logverzeichnis (Windows only)*: Öffnet das Verzeichnis, in welchem alle Dateien gespeichert werden sollen. Dies wird über einen Windowsbefehl implementiert, so dass dies ausschließlich auf Windows-Betriebssystemen funktioniert.
  - **? (Hilfe)**
    - *Hilfe*: Öffnet ein Hilfefenster, das die Vorgehensweise bei der Benutzung des Crawler und der Konvertierung von PDF-Dokumenten beschreibt.
    - *About*: Zeigt das Impressum an.
- **Tabs**
  - **Start**: Begrüßungsnachricht und kurze Beschreibung der Applikation.

- **CiteSeer**: Beinhaltet die Oberfläche zum Erschließen von CiteSeer-Dokumenten.
- **Excel/CSV**: Ist nicht implementiert und daher standardmäßig inaktiv. Sollte dazu dienen, URLs aus einer Excel- oder CSV-Datei herauszulesen, um diese Internetsites anschließend herunterzuladen. Es wurde bereits gestaltet und mit Eventverarbeitung implementiert, jedoch fehlt jegliche Funktionalität.
- **Datenbank**: Ist nicht gestaltet oder implementiert. In diesem Tab sollte die Möglichkeit gegeben werden, URLs aus einer Datenbank auszulesen, um diese Sites anschließend herunterzuladen.

Einige Tabs (Karteireiter) beinhaltet ein Fortschritts-Fenster, in welches jegliche Systemausgaben über den Befehl *System.out.println* übergeben werden, je nachdem welches Tab aktiv ist.

2. Erschließung von CiteSeer-Dokumenten

Zum Download von Dokumenten werden zunächst die allgemeine Vorgehensweise von MyCrawler und anschließend die genaue Benutzung der Applikation beschrieben.

2.1 Vorgehensweise des Crawler

Auf der Internetseite http://www.citeseer.com findet man eine Standard-Suchoberfläche vor. Gibt man nun Suchbegriffe ein um wissenschaftliche Veröffentlichungen zu finden, erscheinen zu diesen Suchbegriffen Treffer auf einer Seite (meisten 20 pro Seite), wobei es mehrere Trefferseiten geben kann.

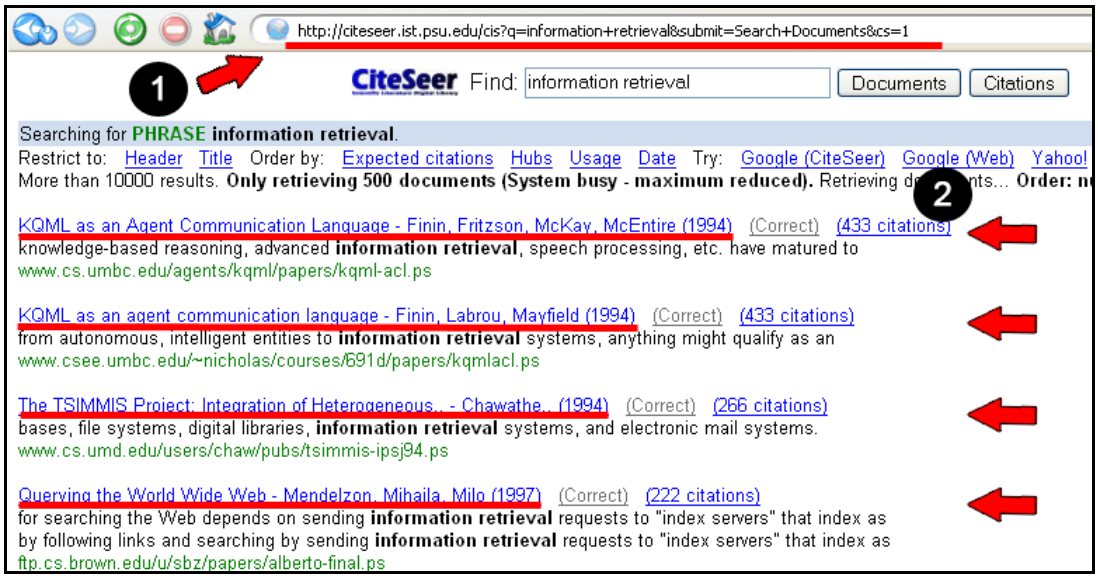


Abbildung 1: Seite mit den Treffern der Suchanfrage.

MyCrawler bietet eine ähnliche Eingabemaske für Suchbegriffe und generiert automatisch die Anfrage URL (Abbildung 1:1). Anschließend wird jede URL jedes Treffers eingelesen (Abbildung 1:2). Die Besonderheit hierbei liegt, dass diese URLs immer auf .html enden und somit die PostScript-Dateien noch nicht direkt zugänglich sind. Zwar steht unter jedem Treffer ein Link zu der PostScript- oder PDF-Datei, allerdings kann es dort vorkommen, dass aufgrund der Länge des Links dieser unvollständig dargestellt wird, um aus Platzgründen auf die Seite zu passen und keinen Zeilenumbruch hervorzurufen. Deswegen werden die einzelnen auf .html-endenden Seiten gespeichert und in die Datei *log.txt* gespeichert. Die bezeichneten Seiten beinhalten Zusatzinformationen zu den einzelnen Veröffentlichungen, sowie die Links zum Download der Papers.

Da es pro Suchbegriff tendenziell mehrere Treffer gibt, jedoch nur 20 pro Seite angezeigt werden, wird diese Vorgehensweise mehrmals wiederholt, allerdings wird als Start-URL immer wieder eine neue genommen. Diese neue URL führt zur nächsten Trefferseite und befindet sich am Ende jeder Seite mit dem Text „Next 20“ (siehe Abbildung 2).

Abbildung 2: Der "Next 20"-Link am Ende jeder Seite.

Anschließend werden wieder alle auf .html-endenden Links gespeichert und in *log.txt* geschrieben. Dieses Prozedere wiederholt sich so oft, bis entweder eine vom Benutzer angegebene maximale Anzahl an zu erschließenden Links erreicht wurde, oder es keine weiteren „Next 20“-Links gibt, womit die letzte Trefferseite erreicht ist.

Bisher wurden also nur die Links der Treffer gesammelt, die auf eine eigene Seite verweisen, jedoch noch nicht auf die Dokumente selbst. Demzufolge ist es nötig, jeden einzelnen dieser Trefferlinks zu verfolgen, um an die URLs der PostScript-Dokumente zu gelangen. Um nun diese URLs zu sammeln, wird jeder einzelne Link weiterverfolgt. Die nun folgende Seite auf der zweiten Ebene ist bei allen Veröffentlichungen, die auf CiteSeer gelistet sind, standardisiert gleich aufgebaut. Sie beinhaltet Titel, Autor, Erscheinungsjahr, Zugriffstatistiken und vieles mehr. Oben rechts auf den Seiten befinden sich immer die Links zum Download des Artikels im gewünschten Format (siehe Abbildung 3).

MyCrawler holt sich nun aus dem Speicher einen Link der ersten Trefferseite, um auf die zweite Ebene zu gelangen. Dort wird unter anderem der Link der PostScript Datei herausgelesen und in der Datei *pdfs.txt* gespeichert (warum die Benennung *pdfs.txt* ist, obwohl PostScript-Dateien erschlossen werden, wird später genauer erklärt).

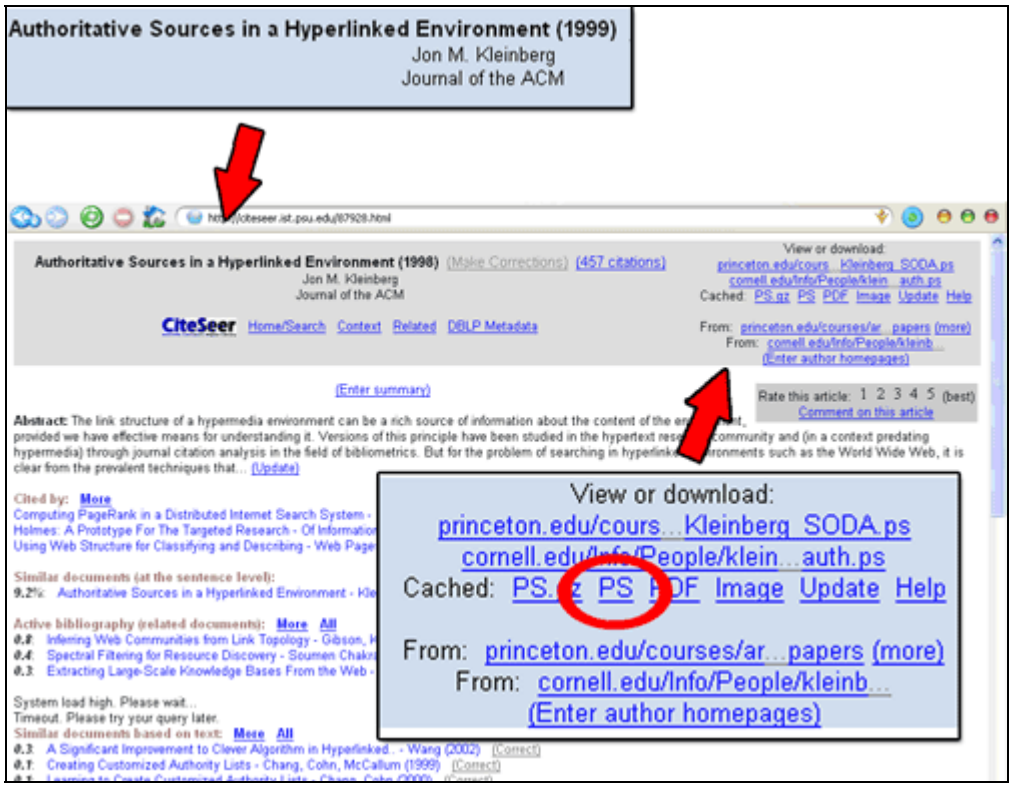


Abbildung 3: Seiten der zweiten Ebene mit detaillierteren Informationen.

Zusätzlich zu diesem Link werden einzelne Informationen aus den BibTex-Informationen extrahiert. BibTeX ist ein Programm zur Erstellung von Literaturangaben und -verzeichnissen in TeX- oder LaTeX-Dokumenten. Alle bekannten Angaben über ein Werk (Buch, Aufsatz, Webseite, etc.) werden in einer bestimmten Syntax notiert. Der Aufbau der BibTeX-Informationen (siehe Abbildung 4) vereinfachte das Parsen nach Titel, Autor, Erscheinungsjahr, URL etc., da diese Informationen dadurch gebündelt weiter unten auf jeder der Seiten zu finden ist. Diese Informationen werden in einer CSV-Datei namens *collection_Suchbegriffe.csv* gespeichert.

Nachdem nun von allen Seiten diese Informationen und Links gesammelt und in eine Datei geschrieben wurden, kann MyCrawler als nächstes mit dem Download der PostScript-Dateien beginnen. Hierzu werden alle Links aus der *pdfs.txt* Datei nacheinander ausgelesen und heruntergeladen. Der Vorteil dabei ist, dass nach Abbruch des Downloads dieser an einer beliebigen Stelle wieder fortgesetzt werden kann. Hierzu muss man einfach die bereits benutzten Links manuelle aus der Datei löschen und den Downloadvorgang erneut starten. Diese Funktion wurde implementiert, weil die Verbindungen zu dem CiteSeer-Server sehr oft

zu langsam waren, so dass es sehr viele Downloadabbrüche gab. Diese Funktion soll komplette Neustarts der Erschließung vermeiden.

Abbildung 4: BibTex Informationen auf der Seite der zweiten Ebene.

2.2 Seiten erschließen

Um Dateien von CiteSeer herunterzuladen, wechselt man in das CiteSeer-Tab.

Abbildung 5: MyCrawler Eingabemaske für CiteSeer.

Zunächst wählt man den Speicherort, an welchem alle Dateien gespeichert werden sollen (Abbildung 5:1). Anschließend kann man festlegen, wie viele Trefferseiten erschlossen werden sollen. Nun kann man in die Suchmaske die Suchbegriffe eingeben. Hierbei sollte darauf geachtet werden, dass es keine unnötigen Leerzeichen vor dem ersten und nach dem letzten Suchbegriff geben darf, da sonst die Generierung der URLs Fehler enthält. Anschließend klickt man auf den Button „Anlegen“ (Abbildung 5:2), welcher einen Ordner im vorher gewählten Speicherordner anlegt, wobei dieser Ordner automatisch den Namen der Suchbegriffe erhält. Gleichzeitig werden in dem angelegten Ordner die Unterordner PS und TEXT angelegt, in denen Später die PostScript-Dateien und die konvertierten Textdokumente gespeichert werden. Als nächstes kann die Art der Sortierung der Treffer festgelegt werden. Wie bei der CiteSeer-Eingabemaske auch, kann bei der Ansicht der Treffer zwischen verschiedenen Sortierungen gewählt werden: Standardsortierung (Anzahl der Zitierungen), nach Expected Citations (den zu erwartenden Zitierungen), nach Usage (Gebrauch der Artikel), nach Hubs (Artikel, die eine Vielzahl an oft zitierten Artikeln anführen) und nach Date (Datum).

Aufgrund der Informationen der Sortierung und der eingegebenen Suchbegriffe generiert CiteSeer automatisch die Start-URL, welche auch bei der Verwendung der Eingabemaske auf CiteSeer.com generiert wird:

- **Standard (Number of Citations)**:

  http://citeseer.ist.psu.edu/cis?q=**SUCHBEGRIFFE**&submit=Search+Documents&cs=1

- **ExpectedCitations:**

  http://citeseer.ist.psu.edu/cs?q=**SUCHBEGRIFFE**&cs=1&submit=Search+Documents&af=Any&ao=Expected+Citations&am=20

- **Usage**:

  http://citeseer.ist.psu.edu/cs?q=**SUCHBEGRIFFE**&cs=1&submit=Search+Documents&af=Any&ao=Usage&am=20

- **Hubs**:

  http://citeseer.ist.psu.edu/cs?q=**SUCHBEGRIFFE**&cs=1&submit=Search+Documents&af=Any&ao=Introductory&am=20

- **Date**:

http://citeseer.ist.psu.edu/cs?q=**SUCHBEGRIFFE**&cs=1&submit=Search+Documents&af=Any&ao=Date&am=20

Die Suchbegriffe werden in den URLs mit einem + getrennt aneinander gefügt.

Das nun folgende Erschließen aller Informationen, URLs, etc. ist in drei Schritte aufgeteilt:

Schritt 1:

Um nun die Links der Trefferseiten zu erfassen, startet man den ersten Crawlingvorgang mit „Go“ (Abbildung 5:3). Nach Abschluss dieses Vorgangs wurde die Datei *logs.txt* geschrieben.

Schritt 2:

Dieser Schritt lässt sich nur ausführen, wenn Schritt 1 ausgeführt wurde. Um nun die Informationen und URLs der einzelnen Seiten der zweiten Ebene zu erhalten, klickt man auf „Sammeln“. Dieser Vorgang kann sehr lange dauern. Ist dieser Prozess abgeschlossen, so existieren im Speicherordner die Dateien *collection_Suchbegriffe.csv* und *pdfs.txt*.

Schritt 3:

Der Downloadvorgang wird nun durch „Download“ gestartet. Hierbei werden die URLs der einzelnen PostScript-Dateien aus der Datei *pdfs.txt* nacheinander genommen und heruntergeladen. Sollte es aufgrund der oftmals sehr schlechten Internetverbindung zu CiteSeer.com einen Abbruch geben, so könnte der Downloadvorgang an der abgebrochenen Stelle fortgesetzt werden, indem man aus der Datei pdfs.txt die bereits erschlossenen URLs manuell herauslöscht. Wichtig: Die Datei muss im gleichen Verzeichnis bleiben und darf nicht umbenannt werden.

Dieser Schritt kann unabhängig von Schritt 1 und Schritt 2 gestartet werden. Voraussetzung ist jedoch, dass es einen angelegten Ordner im Speicherpfad gibt und in diesem eine Datei *pdfs.txt* mit den herunterzuladenden Links existiert.

Nach Abschluss aller Schritte sieht eine Ordnerstruktur folgendermaßen aus:

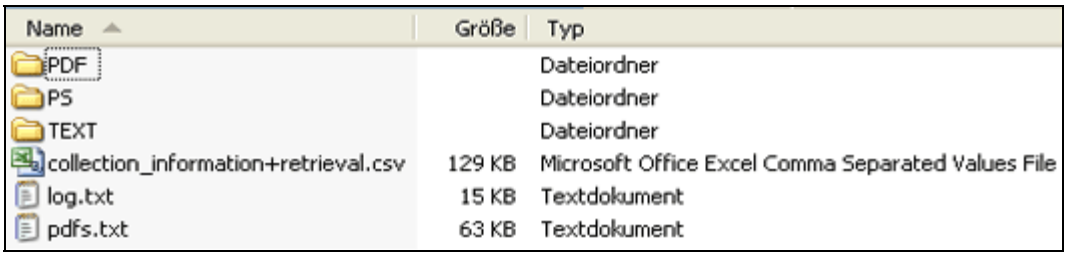


Abbildung 6: Fertige Ordnerstruktur nach Abschluss aller Schritte.

3. Konvertierung von PDF-Dokumenten

Da MyCrawler erst für die Erschließung von PDF-Dateien konzipiert wurde, wurde eine Funktion eingefügt, die die Konvertierung von PDF-Dokumenten in reine Textdokumente vornehmen kann. Da nun aber PostScript-Dateien heruntergeladen werden, müssen diese erst in PDF-Dateien konvertiert werden. Hierzu wurde das Tool Jaws PDFCreator v3.3[11] verwendet, welches mehrere PostScript-Dokumente parallel konvertieren kann.

Um eine Konvertierung in Textdokumente durchzuführen, wechselt man in das PDF-Tab (Abbildung 7). Für diese Funktionalität wurde die Java-Klassenbibliothek PDFBox Version 0.6.7a[12] implementiert.

[11] http://www.jawspdf.com

[12] http://www.pdfbox.org

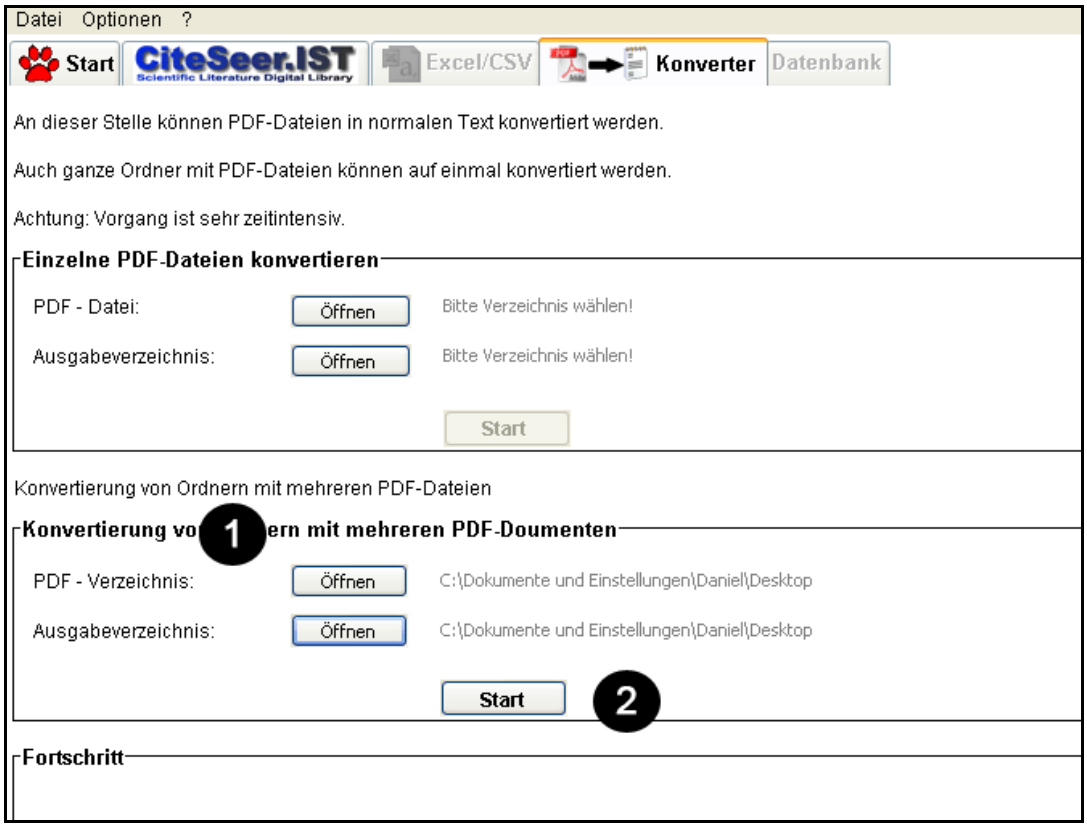


Abbildung 7: PDF-Tab zur Konvertierung in Textdateien.

Nachdem alle PostScript-Dateien in PDF-Dateien konvertiert wurden, kann man nun das Verzeichnis auswählen, in welchem alle PDF-Dateien enthalten sind, die konvertiert werden sollen (Abbildung 7:1). Anschließend wählt man ein Verzeichnis, in welches die fertigen Textdateien gespeichert werden sollen. Hierbei bietet sich der bereits angelegte Ordner TEXT an. Anschließend beginnt die Konvertierung mit „Starten“ (Abbildung 7:2). Es gibt die Möglichkeiten die Konvertierung für einzelne Dokumente oder für ganze Ordner mit PDF-Dokumenten durchzuführen. Analog zur gerade beschriebenen Vorgehensweise für die Konvertierung ganzer Ordner funktioniert die Konvertierung einzelner Dateien.

4. Probleme und Verbesserungsmöglichkeiten

MyCrawler konnte erfolgreich zu Erschließung eines Korpus eingesetzt werden. Jedoch gibt es einige funktionelle Auffälligkeiten, die bei einer Überarbeitung weiterhin berücksichtigt werden könnten.

Verbindungsprobleme zu CiteSeer.com

Bei mehreren Durchläufen hat sich heraus gestellt, dass die Internetverbindung zu CiteSeer.com oftmals unzureichend war, so dass laufenden Prozeduren zum Abbruch gezwungen wurden. Dies kann auf ein Server-Problem von CiteSeer deuten, da auch die Internetseite von CiteSeer sehr oft nicht erreichbar ist bzw. Suchanfragen sehr lange dauern. Da das Erschließen der Informationen, Links und Dokumente ein sehr zeitaufwendiger Prozess ist, wurde der Downloadvorgang so programmiert, dass die URLs, die vorher gesammelt wurden, direkt aus einer Datei gelesen werden und die Dokumente unabhängig vom Fortschritt des Crawler oder von Abbrüchen heruntergeladen werden können.

PDF-Konvertierungsprobleme

Bei der Konvertierung von einem Ordner mit einer Vielzahl an PDF-Dokumenten auf, dass ab 20 Dokumenten die Prozedur sehr lange dauert. Dies kann darauf zurückgeführt werden, dass die PDF-Dokumente teilweise sehr groß sind. Allerdings wird aus Java heraus die PDFBox via Batchbefehl als Thread aufgerufen, was dazu führt, dass alle Prozesse gleichzeitig starten und somit Rechenzeit und Arbeitsspeicher gleichzeitig beanspruchen.

Die Konvertierung funktioniert, jedoch wurde aufgrund der großen Anzahl an Dokumenten und der Größe der PDF-Dateien selbst das Programm PDF2Text v3.0[13] verwendet. Der Vorteil ist, dass PDF2Text auch ganze Ordner konvertieren kann und dabei iterativ vorgeht, wodurch die Rechenzeit erheblich verkürzt wird. Allerdings steht PDF2Text nicht als Java-Klassenbibliothek zur Verfügung.

Ein weiteres Problem bei der PDF-Konvertierung sind fehlerhafte PDF-Dateien, die nach einer Konvertierung in Textdokumente nur Sonderzeichen beinhalten. Dies kann darauf zurückgeführt werden, dass entweder Dokumente gescannt wurden und nicht über einen Texteditor als PDF gespeichert wurden, oder dass die serverseitige Konvertierung von PostScript-Dateien auf CiteSeer.com Probleme bereitet. Die von CiteSeer zur Verfügung

[13] http://www.verypdf.com/pdf2txt/pdf2txt.htm

gestellten PDF-Dokumente befinden sich unter anderem im Cache. Da CiteSeer mehrere Formate zur Verfügung stellt um die Publikationen für Benutzer zugänglich zu machen, kann es sein, dass bei der serverseitigen Konvertierung von PostScript-Dateien Fehler aufgetreten sind. Die konvertierten PDF-Dokumente lassen sich zwar mit dem Adobe Acrobat Reader anzeigen und auch ausdrucken, allerdings funktionierte oftmals eine Konvertierung in eine Textdatei mit unterschiedlichen Tools (Adobe Acrobat Reader, PDFBox, PDF2Text) nicht, so dass bei der Ausgabe nur Sonderzeichen in der Datei enthalten waren. Bei darauf folgenden Tests wurde festgestellt, dass die Fehlerquote deutlich geringer wurde, wenn man nicht die PDF-Dokumente, sondern PostScript-Dokumente herunterlädt. Aus diesem Grund sind viele Bestandteile von MyCrawler auf PDF-Dateien ausgerichtet gewesen und wurden für die Verwendung mit PostScript-Dateien teilweise nur abgeändert, so dass es vorkommen kann, dass Benennungen immer noch PDF beinhalten, aber eine Verarbeitung von PostScript-Dateien durchführen[14].

Zusätzliche Funktionen

Die bereits angesprochene Verwendung von Excel-Dateien oder einer Datenbank zum Herunterladen von ganzen Internetsites könnte nachträglich implementiert werden. Auch könnte ein Schritt bei der Konvertierung in Textdateien wegfallen, wenn man die PostScript-Dateien direkt in Textdateien konvertiert, ohne zuerst PDF-Dokumente zu erzeugen.

Weitere Hinweise

Die korrekte Funktionsweise von MyCrawler könnte durch Änderungen auf CiteSeer.com beeinträchtigt werden. Zum Beispiel hängen die URLs, die nach Eingabe der Suchbegriffe generiert werden, davon ab, ob CiteSeer.com diese ändert oder nicht. Auch wenn das Design von CiteSeer.com geändert wird können Probleme auftreten, da dadurch der HTML-Code, welcher nach den Links und Stichworten geparst wird, andere Tags etc. aufweisen könnte, so dass die festgelegten Links und Stichwörter andere Bezeichnungen enthalten oder an einer anderen Stelle im Code auftreten. Diese Fehler müssten im Quellcode angepasst werden.

[14] Deswegen heißt pdfs.txt auch nicht ps.txt, obwohl dies zutreffender wäre.

Danksagungen

Ich möchte mich ganz herzlich bei folgenden Personen für Ihre Unterstützung und Hilfsbereitschaft bedanken:

Kerstin Bischoff, Dorothee Fesel, Nina Kummer, Jan-Hendrik Scharmann, Sarah Risse, Niels Jensen, Jens Schärdel, Christian „Bob“ Reiss, Dominik Giesa, Robert Strötgen, Johanna Völker vom AIFB Karlsruhe, Paul Buitelaar und Alex Schutz vom DFKI Saarbrücken.

Eigenständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Außerdem versichere ich, dass die Arbeit noch nicht veröffentlicht oder in einem anderen Prüfungsverfahren als Prüfungsleistung vorgelegt wurde.

Hildesheim, im März 2005

_____